Betriebsanleitung

XCx 400 / XCx 800

Betriebsanleitung XCx 400 / XCx 800 • Version 05/13
Artikel-Nr. R4.322.2480.0 (322 385 92)

schleicher
control systems

Zielgruppe

Die Betriebsanleitung ist für geschulte Fachkräfte ausgelegt. Es werden besondere Anforderungen an die Auswahl und Ausbildung des Personals gestellt, die mit dem Automatisierungssystem umgehen. Als Personen kommen z.B. Elektrofachkräfte und Elektroingenieure in Frage, die entsprechend geschult sind (siehe auch Sicherheitshinweise "Personalauswahl und -qualifikation").

Gültigkeit der Betriebsanleitung

ab Software Version V11.21/0

Vorgängerversion der Betriebsanleitung

05/11 08/11 10/11 04/12

Bezugsmöglichkeiten für Betriebsanleitungen

Betriebsanleitungen können kostenlos vom Internet:
http://www.schleicher-electronic.com
geladen, oder unter Angabe der Artikel-Nr. bestellt werden bei:

SCHLEICHER Electronic
GmbH & Co. KG
Pichelswerderstraße 3-5
D-13597 Berlin

Zusätzliche Dokumentationen

Siehe Seite 10

Copyright by

SCHLEICHER Electronic
GmbH & Co. KG
Pichelswerderstraße 3-5
D-13597 Berlin

Telefon +49 30 33005-0
Telefax +49 30 33005-378
Hotline +49 30 33005-304
Internet http://www.schleicher-electronic.com
E-Mail info@schleicher-electronic.com

Änderungen und Irrtum vorbehalten

Inhaltsverzeichnis

1 Sicherheitshinweise 7
1.1 Bestimmungsgemäße Verwendung 7
1.2 Personalauswahl und -qualifikation 8
1.3 Projektierung, Programmierung, Installation, Inbetriebnahme und Betrieb 8
1.4 Gefahren durch elektrische Energie 9
1.5 Wartung und Instandhaltung 9
1.6 Umgang mit verbrauchten Batterien 9
2 Zusätzliche Betriebsanleitungen 10
3 Systemübersicht 11
3.1 Steuerungsaufbau 12
3.2 Montage 13
4 Steuereinheit XCx 400 / XCx 800 14
4.1.1 Frontansicht XCx 800 (Schnittstellen, Bedienelemente, Anzeigen) 14
4.1.2 Frontansicht XCx 400 (Schnittstellen, Bedienelemente, Anzeigen) 15
4.1.3 X1, X2 – sercos III-Schnittstellen (nur XCx800) 16
4.1.4 X3 – Ethernet-Schnittstelle 16
4.1.5 X4 – USB 2.0-Host Schnittstelle 16
4.1.6 X5 – USB Device Schnittstelle 17
4.1.7 X6 – CAN-Schnittstelle 17
4.1.8 X7 – RS 232 / RS 422 / RS 485-Schnittstellen 18
4.1.9 X8 – Betriebsspannung / Betriebsbereitrelais 18
4.1.10 X9 – Standard I/Os 24V DC 19
4.1.11 X10 – High-Speed Eingänge 5V DC 19
4.1.12 X11 – High-Speed Ausgänge 5V DC 19
4.1.13 LED-Anzeigen Steuerung 21
4.2 LED-Anzeigen Ethernet 22
4.2.1 LED-Anzeigen sercos III 22
4.2.2 Betriebsartenschalter 23
4.2.3 Reset-Taster 23
4.2.4 Display nur XCx 800 23
4.3 Technische Daten Steuereinheit XCx 400 / XCx 800 24
4.4 Bestellangaben XCx 400 / 800 27
5 Erweiterungsmodule XCx und Promodul-U 28
6 Inbetriebnahme 30
6.1 Installation von MULTIPROG, OPC-Server und Add-Ons 30
6.2 Systemvoraussetzungen 30
6.2.1 MULTIPROG installieren 31
6.2.2 OPC-Server installieren 32
6.2.3 Add-Ons installieren 33
6.2.4 Schleicher Dialog installieren 34
6.3 Inbetriebnahme der Netzwerk-Schnittstelle 35
6.3.1 Kommunikationsschema 35
6.4 Inbetriebnahme der Ethernet-Schnittstelle 36
6.4.1 Einstellen der IP-Adresse des Service-PC 36
6.4.2 Ethernet-Verbindung herstellen und prüfen 36
6.5 Ändern der IP-Adresse der XCx 37
6.5.1 Kommunikation zur Programmierung mit MULTIPROG 38
6.6 Erste Schritte mit MULTIPROG 41
6.6.1 MULTIPROG starten, neues Projekt öffnen und speichern 41
6.6.2 Ein Projekt kompilieren und zur XCx senden 43
6.7 Einfügen des Koppelspeichers 45
6.7.1 Zugriff auf den Koppelspeicher 46
6.7.2 Hinweise zur Auswahl der Koppelspeicher-Version 47
6.8 Zugriff auf die I/O-Ebene 49
6.8.1 Die Auswahl im Dialogfenster XRIO-, XUIO- bzw. XFIO-Konfiguration 50
6.8.2 Tabelle der RIO-Modultypen und Modulklassen 64
6.9 SPS-Applikationsbeispiele für XCx400 / XCx800 I/O Schnittstellen 65

6.9.1 X9, X10, X11 Bitbelegung: 65
6.9.2 schnelle Ausgänge 0-2 / schnelle Eingänge 0-3: 65
6.9.3 Inkrementalwertgeber 67
6.9.4 Absolutwertgeber 69
6.9.5 PWM1/2 mit den schnellen Ausgängen 0/1 72
6.9.6 SPS-Beispiel für Interrupt mit den Eingängen der XCx 400 / XCx 800 73
6.10 CANopen für dezentrale I/O 78
6.10.1 Spezifikationen 78
6.10.2 Anschlussprinzip und Verkabelung 79
6.10.3 Einstellungen am I/O-Modul RIO 8 I/O CANopen 80
6.10.4 Deklaration des I/O-Treibers für CANopen 80
6.10.5 Deklaration von Netzwerkvariablen in MULTIPROG 82
6.10.6 CANopen-Konfiguration mit "Schleicher CANopen Konfiguration" 83
6.10.7 CANopen-Konfiguration mit "ProCANopen" 84
6.10.8 Konfiguration der sercos III I/O-Geräte 90
6.11 Der Webserver 91
6.11.1 Allgemeine Funktionen und Konzept 91
6.11.2 Aktivierung des Webservers 91
6.11.3 Visualisierungsapplikation 91
6.11.4 Deklaration von Variablen zur Visualisierung 92
6.11.5 Browser / Komponenten 92
6.12 Allgemeine Hinweise zur Inbetriebnahme 93
7 Bedienung der XCx mit dem Schleicher Dialog 94
7.1 Übersicht 94
7.2 Vorbereitung des OPC-Servers 95
7.2.1 Deklaration der OPC-Variablen im SPS-Projekt mit MULTIPROG 95
7.2.2 Konfiguration des OPC-Servers 96
7.2.3 Test des OPC-Servers 98
7.3 Schleicher Dialog 99
7.3.1 Schleicher Dialog Konfigurieren 100
7.3.2 Aufbau der Bedienoberfläche 101
7.3.3 Schleicher Dialog SPS/CNC 102
7.3.4 Aufruf von Activ-Error-Buffer und Log-Book 105
8 Die SPS 106
8.1.1 Programmierung 106
8.1.2 Der Koppelspeicher 106
8.2 SPS-Betriebszustände und Startverhalten 107
8.2.1 Betriebszustände 107
8.2.2 Wechseln der Betriebszustände mit MULTIPROG 108
8.3 Systemvariablen 110
8.4 Bibliotheken und Funktionsbausteine in MULTIPROG 111
8.4.1 Hinweis zu den Variablendeklarationen der Beispielprogramme von FBs 112
8.4.2 Bibliothek CANopen_Vxxx 113
8.4.3 Bibliothek CFB_Vxxx 113
8.4.4 Bibliothek CNC_Vxxx 114
8.4.5 Bibliothek Date_Time 115
8.4.6 Bibliothek MC_Vxxx 115
8.4.7 Bibliothek MMI 116
8.4.8 Bibliothek PLC_Vxxx 116
8.4.9 Bibliothek PNS_Vxxx 117
8.4.10 Bibliothek Profibus_Vxxx 117
8.4.11 Bibliothek SchleicherLib_Vxxx 117
8.4.12 Bibliothek Serial 117
8.4.13 Bibliothek XCx7_Vxxx 118
9 Das Multi-Task-System 119
9.1 Übersicht 119
9.2 Anwender-Task 120
9.2.1 Zyklische Task 120
9.2.2 Ereignis-Task 121
9.2.3 System-Tasks 122

9.2.4 Default-Task 124
9.2.5 Anwender-Task-Information 124
9.2.6 Task-Prioritäten 125
9.2.7 Task und Watchdog 126
9.2.8 Task einfügen und Programm zuweisen 127
10 Displayfunktionen der XCx800 129
10.1 Displayanzeige des Betriebszustands 129
10.2 Display Funktionen in Abhängigkeit vom Betriebsartenschalter 130
11 Die RS232-Schnittstelle am Service-PC einrichten 131
11.1 Auslesen der IP-Adresse der XCx über die RS232-Schnittstelle 132
12 Die CNC 133
13 Anhang 134
13.1 Technische Daten aller Module 134
13.2 Zubehör und Software 135
13.3 Warenzeichenvermerke 135
13.4 Abbildungsverzeichnis 136
13.5 Tabellenverzeichnis 138
13.6 Index 140

Darstellungskonventionen

Sicherheits- und Handhabungshinweise werden in dieser Programmier-anleitung durch besondere Kennzeichnungen hervorgehoben:

Warnung!

Bedeutet, dass Personen, das Automatisierungssystem oder eine Sache beschädigt werden kann, wenn die entsprechenden Hinweise nicht eingehalten werden.

Kursivschrift: Hinweise zur Vermeidung der Gefährdung.

Wichtig! oder Hinweis

Hebt eine wichtige Information hervor, die die Handhabung des Automatisierungssystems oder den jeweiligen Teil der Betriebs-anleitung betrifft.

Weitere Objekte werden folgendermaßen dargestellt:

Objekt	Beispiel
Dateinamen	HANDBUCH.DOC
Menüs / Menüpunkte	*Einfügen / Grafik / Aus Datei*
Pfade / Verzeichnisse	*C:\Windows\System*
Hyperlinks	*http://www.schleicher-electronic.com*
Programmlisten	`MaxTsdr_9.6   = 60` `MaxTsdr_93.75 = 60`
Tasten	<Esc> <Enter> (nacheinander drücken) <Strg+Alt+Entf> (gleichzeitig drücken)
Bezeichner der Konfigurationsdaten	Q23
Namen von Variablen	*`mcMem.axSect[n].bContRel`*

1 Sicherheitshinweise

Der im Folgenden verwendete Begriff Automatisierungssysteme umfasst Steuerungen sowie deren Komponenten (Module), andere Teile (wie z.B. Baugruppenträger, Verbindungskabel), Bediengeräte und Software, die für Programmierung, Inbetriebnahme und Betrieb der Steuerungen genutzt wird. Die vorliegende Betriebsanleitung kann nur einen Teil des Automatisierungssystems (z.B. Module) beschreiben.

Die technische Auslegung der Schleicher-Automatisierungssysteme basiert auf der Produktnorm EN 61131-2 (IEC 61131-2) für speicherprogrammierbare Steuerungen. Für die Systeme und Geräte gilt grundsätzlich die CE-Kennzeichnung nach der EMV-Richtlinie 2004/108/EG und sofern zutreffend auch nach der Niederspannungsrichtlinie 2006/95/EG.

Die Maschinenrichtlinie 2006/42/EG ist nicht anzuwenden, da die in der Richtlinie genannten Schutzziele auch von der Niederspannungs- und EMV-Richtlinie abgedeckt werden.

Sind die Schleicher-Automatisierungssysteme Teil der elektrischen Ausrüstung einer Maschine, müssen sie vom Maschinenhersteller in das Verfahren zur Konformitätsbewertung einbezogen werden. Hierzu ist die Norm DIN EN 60204-1 zu beachten (Sicherheit von Maschinen, allgemeine Anforderungen an die elektrische Ausrüstung von Maschinen).

Von den Automatisierungssystemen gehen bei bestimmungsgemäßer Verwendung und ordnungsgemäßer Unterhaltung im Normalfall keine Gefahren in Bezug auf Sachschäden oder für die Gesundheit von Personen aus. Es können jedoch durch angeschlossene Stellelemente wie Motoren, Hydraulikaggregate usw. bei unsachgemäßer Projektierung, Installation, Wartung und Betrieb der gesamten Anlage oder Maschine, durch Nichtbeachten von Anweisungen in dieser Betriebsanleitung und bei Eingriffen durch ungenügend qualifiziertes Personal Gefahren entstehen.

1.1 Bestimmungsgemäße Verwendung

Die Automatisierungssysteme sind nach dem Stand der Technik und den anerkannten sicherheitstechnischen Regeln gebaut. Dennoch können bei ihrer Verwendung Gefahren für Leib und Leben des Benutzers oder Dritter bzw. Beeinträchtigungen von Maschinen, Anlagen oder anderen Sachwerten entstehen.

Das Automatisierungssystem darf nur in technisch einwandfreiem Zustand sowie bestimmungsgemäß, sicherheits- und gefahrenbewusst unter Beachtung der Betriebsanleitung benutzt werden. Der einwandfreie und sichere Betrieb der Steuerung setzt sachgemäßen Transport, sachgerechte Lagerung und Montage sowie sorgfältige Bedienung und Wartung voraus. Insbesondere Störungen, die die Sicherheit beeinträchtigen können, sind umgehend beseitigen zu lassen.

Die Automatisierungssysteme sind ausschließlich zur Steuerung von Maschinen und Anlagen vorgesehen. Eine andere oder darüber hinausgehende Benutzung gilt nicht als bestimmungsgemäß. Für daraus resultierende Schäden haftet der Hersteller nicht.

Zur bestimmungsgemäßen Verwendung der Automatisierungssysteme sind die in dieser Betriebsanleitung beschriebenen Anweisungen zum mechanischen und elektrischen Aufbau, zur Inbetriebnahme und zum Betrieb zu beachten.

1.2 Personalauswahl und -qualifikation

Wichtig!

Alle Projektierungs-, Programmier-, Installations-, Inbetriebnahme-, Betriebs- und Wartungsarbeiten in Verbindung mit dem Automatisierungssystem dürfen nur von geschultem Personal ausgeführt werden (z.B. Elektrofachkräfte, Elektroingenieure). Das Projektierungs- und Programmierpersonal muss mit den Sicherheitskonzepten der Automatisierungstechnik vertraut sein.

Das Bedienpersonal muss im Umgang mit der Steuerung unterwiesen sein und die Bedienungsanweisungen kennen.

Das Installations-, Inbetriebnahme- und Wartungspersonal muss eine Ausbildung besitzen, die zu Eingriffen am Automatisierungssystem berechtigt.

1.3 Projektierung, Programmierung, Installation, Inbetriebnahme und Betrieb

Das Automatisierungssystem ist in seiner Anwendung zumeist Bestandteil größerer Systeme oder Anlagen, in denen Maschinen gesteuert werden. Bei Projektierung, Installation und Inbetriebnahme der Automatisierungssysteme im Rahmen der Steuerung von Maschinen müssen deshalb durch den Maschinenhersteller und Anwender die Sicherheitsbestimmungen der Maschinenrichtlinie 2006/42/EG beachtet werden. Im spezifischen Einsatzfall geltende nationale Unfallverhütungsvorschriften wie z.B. BGV A 3 (VBG 4).

Alle sicherheitstechnischen Vorrichtungen der gesteuerten Maschine sind so auszuführen, dass sie unabhängig von der Steuerung funktionieren. Not-Aus-Einrichtungen müssen in allen Betriebsarten der Steuerung wirksam bleiben. Im Not-Aus-Fall müssen die Versorgungsspannungen aller von der Steuerung angesteuerten Schaltelemente in einen sicheren Zustand gebracht werden.

Es sind Vorkehrungen zu treffen, dass nach Spannungseinbrüchen und -ausfällen ein unterbrochenes Steuerungsprogramm ordnungsgemäß wieder aufgenommen werden kann. Dabei dürfen auch kurzzeitig keine gefährlichen Betriebszustände auftreten. Gegebenenfalls ist Not-Aus zu erzwingen.

Damit ein Leitungsbruch auf der Signalseite nicht zu undefinierten Zuständen in der Steuerung führen kann, sind bei der E/A-Kopplung hard- und softwareseitig entsprechende Sicherheitsvorkehrungen zu treffen. Einrichtungen der Steuerungstechnik und deren Bedienelemente sind so einzubauen, dass sie gegen unbeabsichtigte Betätigung ausreichend geschützt sind.

1.4 Gefahren durch elektrische Energie

Warnung!

Nach Öffnen des Systemschrankes oder nach Entfernen des Gehäuses von Systemkomponenten werden bestimmte Teile des Automatisierungssystems zugänglich, die unter gefährlicher Spannung stehen können.

Die Spannung abschalten, bevor an den Geräten gearbeitet wird. Bei Messungen unter Spannung Kurzschluss vermeiden.

Der Anwender muss dafür sorgen, dass unbefugte und unsachgemäße Eingriffe unterbunden werden (z.B. verschlossener Schaltschrank).

Das Personal muss gründlich mit allen Gefahrenquellen und Maßnahmen zur Inbetriebnahme und Wartung gemäß den Angaben in der Betriebsanleitung vertraut sein.

1.5 Wartung und Instandhaltung

Werden Mess- oder Prüfarbeiten am aktiven Gerät erforderlich, dann sind die Festlegungen und Durchführungsanweisungen der nationalen Unfallverhütungsvorschriften, wie z.B. BGV A 3 (VBG 4), zu beachten. Es ist geeignetes Elektrowerkzeug zu verwenden.

Reparaturen an Steuerungskomponenten dürfen nur von autorisierten Reparaturstellen vorgenommen werden. Unbefugtes Öffnen und unsachgemäße Eingriffe oder Reparaturen können zu Körperverletzungen oder Sachschäden führen.

Vor Öffnen des Gerätes ist immer die Verbindung zum speisenden Netz zu trennen (Netzstecker ziehen oder Trennschalter öffnen).

Steuerungsmodule dürfen nur im spannungslosen Zustand gewechselt werden. Demontage und Montage sind gemäß den mechanischen Aufbaurichtlinien vorzunehmen.

Beim Auswechseln von Sicherungen dürfen nur Typen verwendet werden, die in den technischen Daten spezifiziert sind.

Beim Austausch von Batterien dürfen nur Typen verwendet werden, die in den technischen Daten spezifiziert sind. Batterien sind in jedem Fall nur als Sondermüll zu entsorgen.

1.6 Umgang mit verbrauchten Batterien

Die in den Automatisierungssystemen verwendeten Batterien sind, nach deren Gebrauchsende, dem Gemeinsamen Rücknahmesystem Batterien (GRS) oder öffentlich-rechtlichen Entsorgungsträgern zuzuführen.

Batterien sollen nur im entladenen Zustand zurückgegeben werden. Der entladene Zustand ist erreicht, wenn eine Funktionsbeeinträchtigung des Gerätes wegen unzureichender Batteriekapazität vorliegt.

Bei nicht vollständig entladenen Batterien muss Vorsorge gegen mögliche Kurzschlüsse getroffen werden. Das kann durch Isolieren der Batteriepole mit Klebestreifen erreicht werden.

2 Zusätzliche Betriebsanleitungen

Wichtig!

Die XCx 400 / XCx 800 sind Mitglied der XCx-Steuerungsfamilie, die auf einem gemeinsamen Soft- und Hardwarekonzept basiert. Zur Ergänzung der vorliegenden Betriebsanleitung müssen daher noch folgende Betriebsanleitungen verwendet werden.

Tabelle 1: Zusätzliche Betriebsanleitungen

Bezeichnung	Artikel-Nr. bzw. Referenz
Zur Inbetriebnahme der Feldbusse	
EMV-Richtlinien deutsch	R4.322.1060.0
Inbetriebnahmehinweise für Feldbussysteme	R4.322.1600.0
sercos III-I/O	R6.322.0770.0
Für die Programmierung der SPS und der CNC	
MULTIPROG Programmiersystem nach IEC 61131-3	MULTIPROG-Handbuch deutsch (Quickstart_MWT.pdf) im Installationspfad von MULTIPROG
CNC-Programmierung XCx und ProNumeric	R4.322.2080.0
Koppelspeicherbelegung der XCx	Online-Hilfe des Softwarepaketes zur XCx
Zur Inbetriebnahme der CNC	
CNC-Inbetriebnahme XCx und ProNumeric	R4.322.2340.0
Getting Started MULTIPROG	R4.322.2460.0
Für die Baugruppenträger, Netzteile und Erweiterungsmodule	
Erweiterungsmodule für XCx	R4.322.2400.0
RIO Erweiterungsmodule	R4.322.1720.0
XRIO Buskoppler RIO EC 2	R5.363.0160.0

Alle Betriebsanleitungen sind als PDF-Dateien auf der Service-CDROM zur XCx verfügbar und können kostenlos von der Website *http://www.schleicher-electronic.com* geladen werden.

3 Systemübersicht

Die XCx 400 / XCx 800 sind Steuereinheiten des Steuerungssystems Promodul-X.

Das System Promodul-X ist eine (evolutionäre) Weiterentwicklung der bewährten Promodul-U Produktreihe und ergänzt diese um neue Module mit leistungsfähigen technischen Merkmalen und Funktionen (vgl. Modul „XSF“).

Die Steuereinheiten der XCx 400 / XCx 800 sind mit einem integrierten Netzteil und einer „intelligenten“ I/O-Karte ausgestattet und können sowohl stand-alone oder auch alternativ zum Anschluss von Erweiterungsmodulen betrieben werden.

Die leistungsfähigere XCx 800 besitzt im Vergleich zur XCx 400 - bei sonst gleicher Ausstattung - zusätzlich ein Display sowie ein SERCOS III Interface.

Sie ist in der Lage, bei höchster Performance und offener, modularer Architektur eine Vielzahl von komplexen Automatisierungsaufgaben und Einsatzszenarien integral abzudecken. Konventionelle Anforderungen wie Steuern, Regeln, Bedienen, Diagnostizieren und Melden werden durch die XCx 400 / XCx 800 auf einer einheitlichen, skalierbaren Plattform bedient.

Applikationsseitig ist die Kompatibilität der SPS- und CNC-Programme gewährleistet.

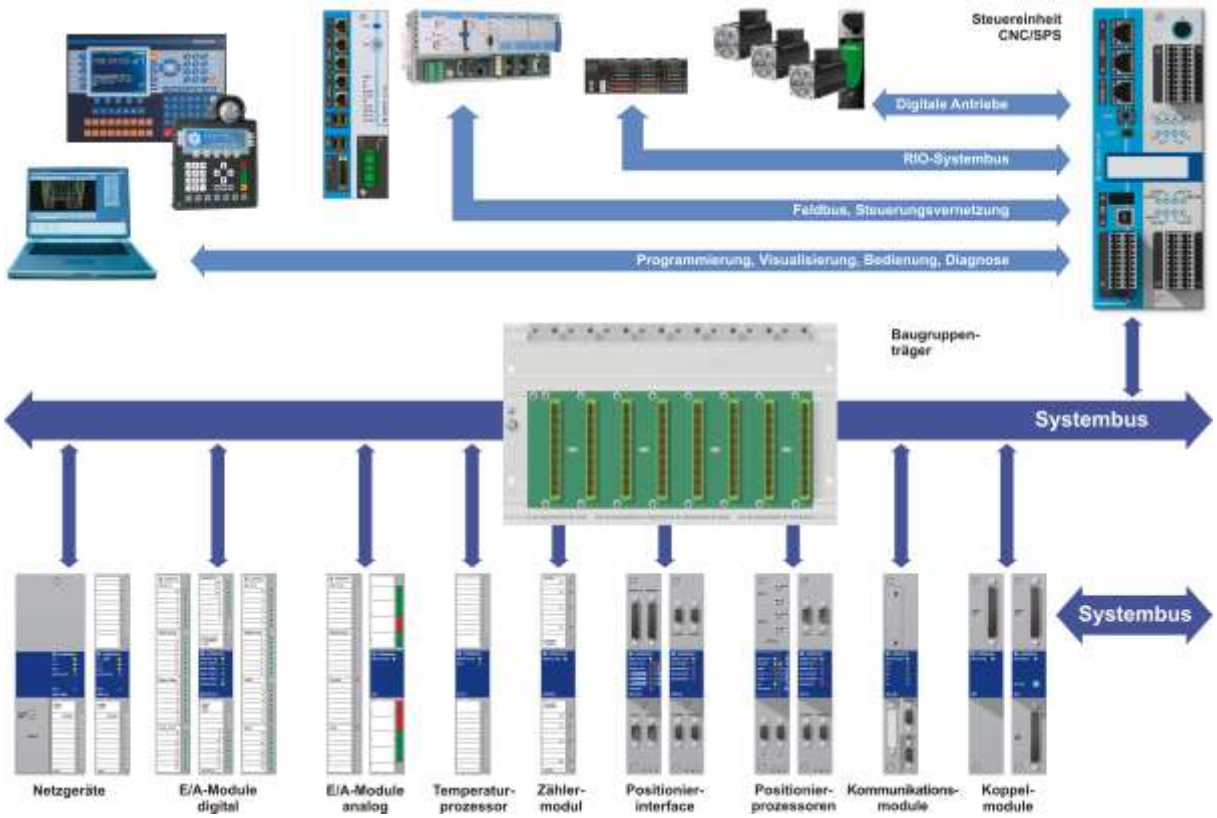


Bild 1: Systemübersicht XCx 400 / 800, Erweiterungsmodule und Peripherie

3.1 Steuerungsaufbau

Das Automatisierungssystem XCx 400 / XCx 800 ist modular aufgebaut, es können bis zu 128 Module auf maximal 8 Baugruppenträgern angeordnet werden.

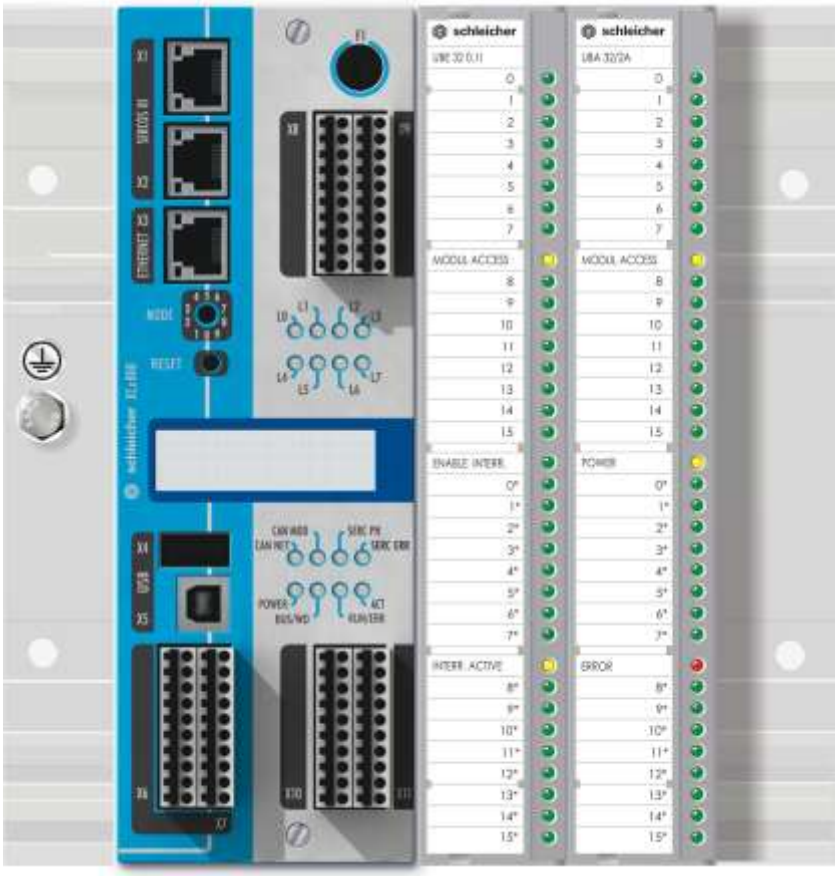
Wichtig!

Der Einbau des Automatisierungssystems muss in geerdeten metallischen Gehäusen (z.B. Schaltschränken) erfolgen. Es sind die in der Dokumentation "EMV-Richtlinien für den Aufbau von Automatisierungsgeräten" (→ Seite 10) dargelegten Vorschriften zu beachten.

Die CPU ist ein Modul im Promodul-U System-Design. Bauhöhe und –tiefe sind entsprechend ausgeführt. Die Steuereinheit mit integriertem Netzteil belegt insgesamt eine Breite von zwei Standard-U-Modulen. Aus dem integrierten Netzteil werden sowohl die CPU selbst wie auch der Rückwandbus und die Erweiterungsmodule auf dem Baugruppenträger versorgt.

Zur Programm- und Datenspeicherung kommen Compact Flash zur Anwendung.

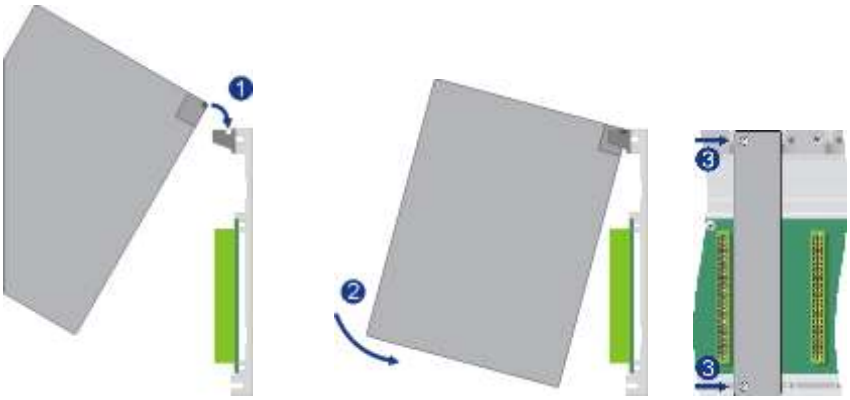
Bild 2:
Aufbau des Gesamtsystems



3.2 Montage

Der Steckplatz der Steuerung auf einem Baugruppenträger der XBT-Reihe befindet sich links von den Erweiterungsmodulen. Diese Steckplatzreihenfolge ist unbedingt einzuhalten!

Bild 3:
Montage der Module auf dem Baugruppenträger



1.
Modul mit den seitlichen Zapfen von oben in den Baugruppenträger einhängen

2.
Modul fest in die Kontaktleiste drücken

3.
Befestigungsschrauben anziehen

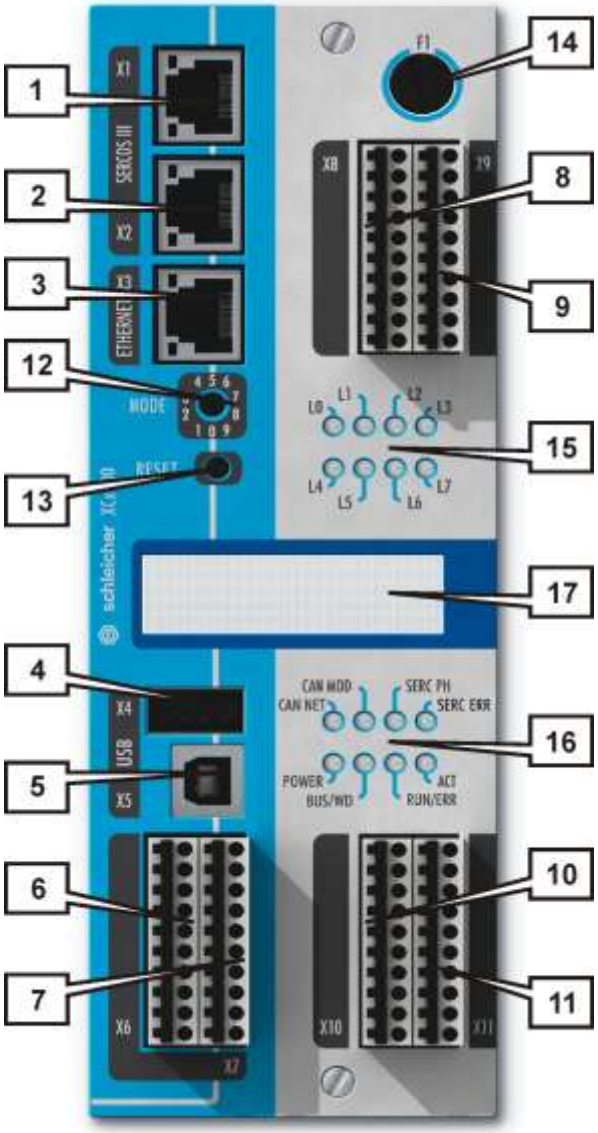
Hinweis

Weitere Hinweise zum Gesamtsystem, zur Bestückung der Baugruppenträger, zur Bemessung der erforderlichen Netzgerätleistung sowie zur Auswahl der Erweiterungsmodule finden Sie in der Betriebsanleitung " Erweiterungsmodule für XCx " (→ Seite 10)

4 Steuereinheit XCx 400 / XCx 800

4.1.1 Frontansicht XCx 800 (Schnittstellen, Bedienelemente, Anzeigen)

Bild 4: Steuereinheit XCx 800



1 / 2 X1, X2
sercos III-Schnittstellen, RJ 45

3 X3
Ethernet-Schnittstelle, RJ 45

4 X4
USB-Host

5 X5
USB-Device

6 X6
CAN-Schnittstelle

7 X7
RS 232 / RS 422 / RS 485

8 X8
Betriebsspannung 24 V, Betriebsbereit

9 X9
Inputs / Outputs 0, 1, 2, 3 (24 V)
Inputs 4, 5, 6, 7 (24 V)

10 X10
Inputs 0, 1, 2, 3 (5 V, differentiell)

11 X11
Outputs 0, 1, 2 (5V, differentiell)

12 MODE
Betriebsartenschalter

13 Reset
Reset-Taster

14 F1
Sicherung

15 L0, L1, …, L7
LED-Anzeigen: Inputs / Outputs 0..7 (24 V)

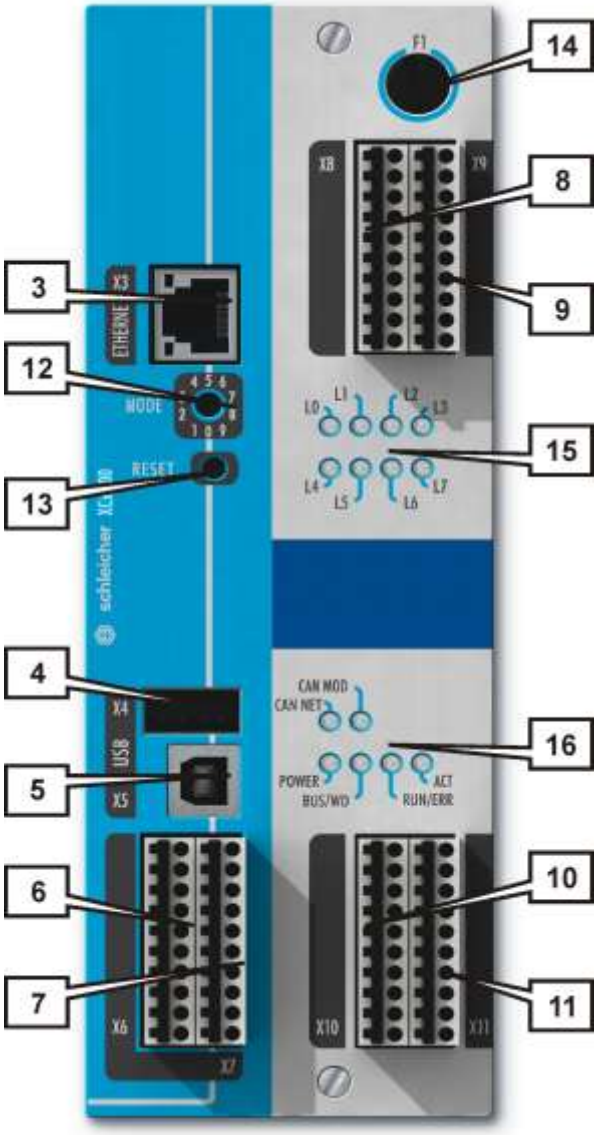
16 CAN NET, CAN MOD, …, ACT
LED-Anzeigen: Controller Status

17 Display
Einfarbiges Grafik-Display mit Backlight

X6	X7	X8	X9	X10	X11
V+	SHLD	24V	24Vext	IN0 (A)	OUT0
CAN_H	TxD	24V	Mext	/IN0 (/A)	/OUT0
DRAIN	RxD	M	I/O0	IN1 (B)	OUT1
CAN_L	GND	M	I/O1	/IN1 (/B)	/OUT1
V-	GND	nc	I/O2	IN2 (N)	OUT2
V+	TD-	nc	I/O3	/IN2 (/N)	/OUT2
CAN_H	TD+	nc	I4	nc	IN3
DRAIN	RD-	Öffner	I5	24VF	/IN3
CAN_L	RD+	Schließer	I6	MF	MF
V-	SHLD	Wurzel	I7	SHLD	SHLD

4.1.2 Frontansicht XCx 400 (Schnittstellen, Bedienelemente, Anzeigen)

Bild 5: Steuereinheit XCx 400



3 X3
Ethernet-Schnittstelle, RJ 45

4 X4
USB-Host

5 X5
USB-Device

6 X6
CAN-Schnittstelle

7 X7
RS 232 / RS 422 / RS 485

8 X8
Betriebsspannung 24 V, Betriebsbereit

9 X9
Inputs / Outputs 0, 1, 2, 3 (24 V)
Inputs 4, 5, 6, 7 (24 V)

10 X10
Inputs 0, 1, 2, 3 (5 V, differentiell)

11 X11
Outputs 0, 1, 2 (5V, differentiell)

12 MODE
Betriebsartenschalter

13 RESET
Reset-Taster

14 F1
Sicherung

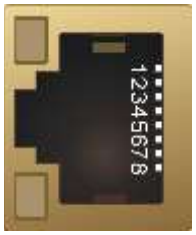
15 L0, L1, …, L7
LED-Anzeigen: Inputs / Outputs 0..7 (24 V)

16 CAN NET, CAN MOD, …, ACT
LED-Anzeigen: Controller Status

X6	X7	X8	X9	X10	X11
V+	SHLD	24V	24Vext	IN0 (A)	OUT0
CAN_H	TxD	24V	Mext	/IN0 (/A)	/OUT0
DRAIN	RxD	M	I/O0	IN1 (B)	OUT1
CAN_L	GND	M	I/O1	/IN1 (/B)	/OUT1
V-	GND	nc	I/O2	IN2 (N)	OUT2
V+	TD-	nc	I/O3	/IN2 (/N)	/OUT2
CAN_H	TD+	nc	I4	nc	IN3
DRAIN	RD-	Öffner	I5	24VF	/IN3
CAN_L	RD+	Schließer	I6	MF	MF
V-	SHLD	Wurzel	I7	SHLD	SHLD

4.1.3 X1, X2 – sercos III-Schnittstellen (nur XCx800)

Tabelle 2: Pinbelegung der sercos III-Schnittstellen X1, X2



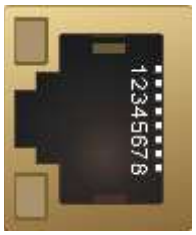
RJ 45 Buchse

Pin	Bezeichnung	Erläuterung
1	TX+	Sendedaten plus
2	TX-	Sendedaten minus
3	RX+	Empfangsdaten plus
4	nc	nicht angeschlossen
5	nc	nicht angeschlossen
6	RX-	Empfangsdaten minus
7	nc	nicht angeschlossen
8	nc	nicht angeschlossen

Die sercos III-Schnittstellen sind nur bei entsprechend ausgestatteten Steuereinheiten aktiv (→ "Bestellangaben XCx 400 / 800", S. 27).

4.1.4 X3 – Ethernet-Schnittstelle

Tabelle 3: Pinbelegung der Ethernet-Schnittstellen X3 (RJ 45)



RJ 45 Buchse

Pin	Bezeichnung	Erläuterung
1	TX+	Sendedaten plus
2	TX-	Sendedaten minus
3	RX+	Empfangsdaten plus
4	nc	nicht angeschlossen
5	nc	nicht angeschlossen
6	RX-	Empfangsdaten minus
7	nc	nicht angeschlossen
8	nc	nicht angeschlossen

4.1.5 X4 – USB 2.0-Host Schnittstelle

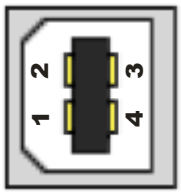
Tabelle 4: Pinbelegung der USB-Schnittstellen X4

USB Buchse Standard A

Pin	Bezeichnung	Erläuterung
1	VCC	+5 V
2	D-	Data minus
3	D+	Data plus
4	GND	Ground

4.1.6 X5 – USB Device Schnittstelle

Tabelle 5: Pinbelegung der USB-Schnittstellen X5

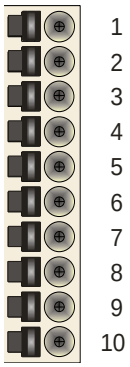


USB Buchse Standard B

Pin	Bezeich-nung	Erläuterung
1	VCC	+5 V
2	D-	Data minus
3	D+	Data plus
4	GND	Ground

4.1.7 X6 – CAN-Schnittstelle

Tabelle 6: Pinbelegung der CAN-Schnittstellen X6



Federkraftan-schluss-Block-klemme 10-polig

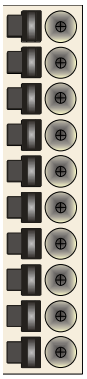
Pin	Bezeich-nung	Erläuterung
1	V+	nicht angeschlossen
2	CAN_H	CAN High
3	DRAIN	Schirmanschluss (optional)
4	CAN_L	CAN Low
5	V-	Ground 0V
6	V+	nicht angeschlossen
7	CAN_H	CAN High
8	DRAIN	Schirmanschluss (optional)
9	CAN_L	CAN Low
10	V-	Ground 0V

Die Pingruppen 1..5 und 6..10 sind parallel geschaltet.

Die Federkraftanschluss-Blockklemmen sind kodiert, um ein Vertauschen der Schnittstellen X6 - X11 zu verhindern.

4.1.8 X7 – RS 232 / RS 422 / RS 485-Schnittstellen

Tabelle 7: Pinbelegung der RS 232 / 422 / 485-Schnittstellen X7



Federkraftan-schluss-Block-klemme 10-polig

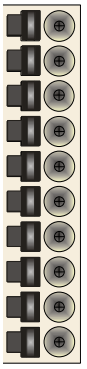
Pin	Bezeich-nung	Erläuterung
1	SHLD	Schirm RS 232
2	TxD	RS 232 Sendedaten
3	RxD	RS 232 Empfangsdaten
4	GND	Masse für RS 232
5	GND	Masse für RS 422 / RS 485
6	TD-	Sendedaten / Sende- und Empfangsdaten
7	TD+	Sendedaten / Sende- und Empfangsdaten
8	RD-	Empfangsdaten / Busabschlusswiderstände
9	RD+	Empfangsdaten / Busabschlusswiderstände
10	SHLD	Schirm RS 422 / RS 485

Die RS-Schnittstellen dienen dem Anschluss von Bedien- und Anzeigegeräten. Bei Benutzung der RS 485-Schnittstelle sind zur Aktivierung der Busabschlusswiderstände RD+ mit TD+ und RD- mit TD- zu verbinden. Die beiden Masse-Pins für RS 232 und RS 485 haben das gleiche Potenzial. PC-Einrichtdaten für RS232 (→ Seite 131).

Die Federkraftanschluss-Blockklemmen sind kodiert, um ein Vertauschen der Schnittstellen X6 - X11 zu verhindern.

4.1.9 X8 – Betriebsspannung / Betriebsbereitrelais

Tabelle 8: Pinbelegung der Betriebsspannung / Betriebsbereitrelais - Schnittstellen X7



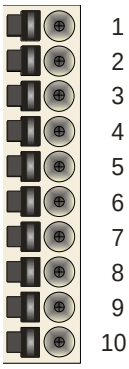
Federkraftan-schluss-Block-klemme 10-polig

Pin	Bezeich-nung	Erläuterung
1	24V	24 V DC Betriebsspannung
2	24V	24 V DC Betriebsspannung
3	M	Masse Betriebsspannung
4	M	Masse Betriebsspannung
5	NC	nicht angeschlossen
6	NC	nicht angeschlossen
7	NC	nicht angeschlossen
8	Öffner	Betriebsbereitkontakt
9	Schließer	Betriebsbereitkontakt
10	Wurzel	Betriebsbereitkontakt

Die Federkraftanschluss-Blockklemmen sind kodiert, um ein Vertauschen der Schnittstellen X6 - X11 zu verhindern.

4.1.10 X9 – Standard I/Os 24V DC

Tabelle 9: Pinbelegung der 24V I/O-Schnittstellen X9



Federkraftan-schluss-Block-klemme 10-polig

Pin	Bezeich-nung	Erläuterung
1	$24V_{ext}$	24 V DC Betriebsspannung für I/Os
2	M_{ext}	Masse Betriebsspannung für I/Os
3	I/O 0	Ein-/Ausgang 0
4	I/O 1	Ein-/Ausgang 1
5	I/O 2	Ein-/Ausgang 2
6	I/O 3	Ein-/Ausgang 3
7	I 4	Eingang 4
8	I 5	Eingang 5
9	I 6	Eingang 6
10	I 7	Eingang 7

Die Federkraftanschluss-Blockklemmen sind kodiert, um ein Vertauschen der Schnittstellen X6 - X11 zu verhindern.

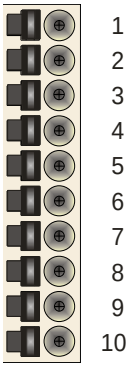
Funktionsbeispiel siehe unter X9, X10, X11 Bitbelegung: Seite 65.

Wichtig!

Die Spannungsversorgung an X9, Pin 1 und 2 versorgt die Steckblockklemmen X9, X10, X11 und muss angeschlossen werden, wenn eine der Federkraftanschluss-Blockklemmen beschaltet wird.

4.1.11 X10 – High-Speed Eingänge 5V DC

Tabelle 10: Pinbelegung der 5V Eingangs - Schnittstellen X10



Federkraftan-schluss-Block-klemme 10-polig

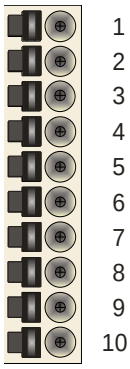
Pin	Bezeich-nung	Erläuterung
1	IN 0 (A)	Eingang 0 (Gebersignal A)
2	/IN 0 (/A)	inv. Eingang 0 (Gebersignal inv. A)
3	IN 1 (B)	Eingang 1 (Gebersignal B)
4	/IN 1 (/B)	inv. Eingang 1 (Gebersignal inv. B)
5	IN 2 (N)	Eingang 2 (Gebersignal N)
6	/IN 2 (/N)	inv. Eingang 2 (Gebersignal inv. N)
7	NC	nicht angeschlossen
8	24VF	gefilterte Betriebsspannung für Geber
9	MF	Masse für gefilterte Betriebsspannung
10	SHLD	Schirm

Die Federkraftanschluss-Blockklemmen sind kodiert, um ein Vertauschen der Schnittstellen X6 - X11 zu verhindern.

Funktionsbeispiel siehe unter Zugriff auf die I/O-Ebene Seite 49.

4.1.12 X11 – High-Speed Ausgänge 5V DC

Tabelle 11: Pinbelegung der 5V Ausgangs - Schnittstellen X11



Federkraftan-schluss-Blockklemme 10-polig

Pin	Bezeich-nung	Erläuterung	Bezeich-nung	Erläuterung
	In / Out		XRIO-Bus	
1	OUT 0	Ausgang 0		
2	/OUT 0	inv. Ausgang 0		
3	OUT 1	Ausgang 1		
4	/OUT 1	inv. Ausgang 1		
5	OUT 2	Ausgang 2	Data Out	XRIO-Data
6	/OUT 2	inv. Ausgang 2	Data /Out	XRIO-Data
7	IN 3	Eingang 3	Data In	XRIO-Data
8	/IN 3	inv. Eingang 3	Data /In	XRIO-Data
9	MF	Masse	M	Masse
10	SHLD	Schirm	SHLD	Schirm

Die Federkraftanschluss-Blockklemmen sind kodiert, um ein Vertauschen der Schnittstellen X6 - X11 zu verhindern.

Funktionsbeispiel siehe unter Zugriff auf die I/O-Ebene Seite 49.

Wichtig!

Die Spannungsversorgung an X9, Pin 1 und 2 versorgt die Steckblockklemmen X9, X10, X11 und muss angeschlossen werden, wenn eine der Federkraftanschluss-Blockklemmen beschaltet wird.

- XRIO

Verdrahtungshinweis zum Anschluss von RIO EC X2 (XRIO) an X11 der XCx 400 / XCx 800:

Tabelle 12: Anschluss der Schnittstelle X11 an den Buskoppler RIO EC X2

XCx 400 / XCx 800		RIO EC X2 (9pol D-Sub)
Klemme X11		X2 IN
Pin 5	<->	Pin 2
Pin 6	<->	Pin 7
Pin 7	<->	Pin 1
Pin 8	<->	Pin 6
Pin 9	<->	Pin 3
Pin10	<->	Kabelschirm

4.1.13 LED-Anzeigen Steuerung

Tabelle 13: LED-Anzeigen auf der Steuerung

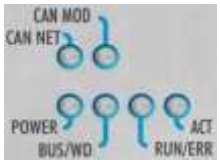
XCx 400

XCx 800

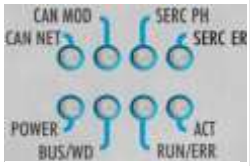
LED-Bezeichnung	Farbe	Zustand	Bedeutung
CAN NET			**Netzwerkzugriffe**
		aus	CAN State Prepared
	grün	dauernd ein	CAN State Operational
		blinkend	CAN State Pre-Operational
	rot	dauernd ein	Bus Off
		blinkend	CAN-Fehler
CAN MOD			**CAN Modulstatus**
	grün	dauernd ein	CAN-Stack initialisiert
		blinkend	ungültige CAN-Konfiguration
	rot	dauernd ein	Steuereinheit nicht bereit oder schwerer Fehler
		blinkend	Fehler in der Steuerung
SERC PH			**SERCOS Phasen**
	rot	dauernd ein	SERCOS Phase 0
	rot	blinkend	SERCOS Phase 1
	gelb	blinkend	SERCOS Phase 2
	grün	blinkend	SERCOS Phase 3
		dauernd ein	SERCOS Phase 4
SERC ERR			**SERCOS Fehler**
	rot	aus	kein Fehler
		dauernd ein	Kommunikationsfehler
		blinkend	Antriebsfehler
Power			**Power**
		aus	Gerät ausgeschaltet
	grün	dauernd ein	Gerät eingeschaltet
BUS/WD			**Buszugriff / Watchdog**
		aus	kein Buszugriff bei SPS-Stopp oder Betriebssystem nicht aktiv
	grün	dauernd ein	Buszugriff in Ordnung
	rot	blinkend	Buszugriffsfehler / Konfigurationsfehler
	rot	dauernd ein	schwerwiegender Fehler oder Betriebssystem nicht aktiv, Watchdog hat angesprochen

(Fortsetzung)

LED-Anzeigen auf der Steuerung XCx 400



XCx 800



RUN/ERR			**CPU-Status**
		aus	CPU defekt oder Betriebssystem nicht aktiv
	Grün / rot	blinkend	im Wechsel blinkend: Betriebssystem startet
	gelb	dauernd ein	CPU läuft, Betriebsspannung in Ordnung, kein Fehler, SPS-Stopp
	grün	dauernd ein	SPS läuft
	gelb	blinkend	SPS läuft, aber Ausgänge sind abgeschaltet (Betriebsbereit-Relais abgefallen)
	rot	blinkend	fataler Fehler Betriebssystem startet nicht
ACT			**IDE (CF) Aktivität**
		aus	kein Zugriff
	grün	blinkend	Zugriff erfolgt

Fehlermeldungen werden im Active-Error-Buffer und im Error-Logbook gespeichert, sie sind mit Fehlernummern und zusätzlichen Angaben gekennzeichnet.
Active-Error-Buffer und Log-Book sind im Schleicher Dialog auf jeder Bedienebene über die Tastenkombination <Ctrl+?> aufrufbar.

4.2 LED-Anzeigen Ethernet

Tabelle 14: LED-Anzeigen an der Ethernet- Buchse (1x RJ45)

Bezeichnung	Farbe	Zustand	Bedeutung
LED 1			Ethernet LED 1 oben (Duplex)
		aus	keine Netzwerkverbindung oder 10 Mbit/s
	grün	dauernd ein	100 Mbit/s Full Duplex Betrieb
LED 2			Ethernet LED 2 unten (Link / Activity / Speed)
		aus	keine Netzwerkverbindung
	gelb	blinkend	Verbindung aktiv

4.2.1 LED-Anzeigen sercos III

Tabelle 15: LED-Anzeigen an den sercos-III-Buchsen (2x RJ45)

Bezeichnung	Farbe	Zustand	Bedeutung
LED 1			Ethernet LED 1 oben (Link)
		aus	keine Netzwerkverbindung
	grün	dauernd ein	Netzwerkverbindung hergestellt
LED 2			Ethernet LED 2 unten (Activity)
		aus	keine Aktivität
	gelb	blinkend	100 MBit/s Verbindung aktiv

4.2.2 Betriebsartenschalter

Der Betriebsartenschalter verfügt über zehn Stellungen. Das SPS-Startverhalten wird mit dem Betriebsartenschalter eingestellt. Es können folgende Varianten ausgewählt werden, mit denen das Hochlaufverhalten der Steuerung bestimmt wird. Weitere Funktionen siehe Seite 130

Tabelle 16: Betriebsartenschalter

Stellung / Bezeichnung	Bedeutung
0	Grundinitialisierung / Diagnose (Start des Echtzeitbetriebssystems im abgesicherten Modus und Rücksetzen des remanenten Datenspeichers, → S. 93)
1 / Prog	Betriebsart Programmierung (SPS-Stopp)
2 / Warm (auch 4..9)	Warmstart der SPS nach IEC 61131-3 (Defaultstellung)
3 / Cold	Kaltstart der SPS nach IEC 61131-3 (Reinitialisierung der Retainvariablen)

Die Display Funktionen in Abhängigkeit vom Betriebsartenschalter sind in Abschnitt 10.2 dargestellt

Die aktuelle Einstellung des Betriebsartenschalters kann im SPS-Programm abgefragt werden (Arbeitsblatt *Global_Variables, PLC_COMMON:* `cmpSwrdPlcRd_lXModeSwitch`).

4.2.3 Reset-Taster

Der Reset-Taster ermöglicht das Rücksetzen des CPU-Moduls:
Die Reset-Funktion ist mit der Stellung des Betriebsartenschalters verknüpft. Ein Reset wird nur in Stellung "0 = Diagnose“ ausgelöst.

Tabelle 17: Reset-Taster

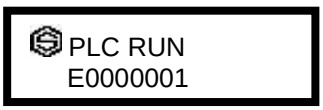
Wichtig!

Wird RESET bei Zugriffen auf die CF Card ausgelöst (LED *ACT* blinkt grün), kann Datenverlust auftreten.

4.2.4 Display nur XCx 800

Zur Anzeige des Systemstatus und erweiterten Diagnose siehe Seite 129.

Bild 6: Display



4.3 Technische Daten Steuereinheit XCx 400 / XCx 800

Weitere technische Daten auf Seite 134

Elektrische Daten		
Versorgungsspannung 24V, M		DC 24 V (19,2 V .. 30 V einschl. 5% Restwelligkeit)
Leistungsaufnahme		max. 75 W
Eingangssicherung		4 A träge
Ausgangsspannung / -strom für U/X-Erweiterungsmodule		DC 24 V / 2 A DC 5 V / 4 A
Ausgangsleistung		DC 24 V / max. 48W DC 5V / max. 20 W Die Gesamtleistung darf 50 W nicht überschreiten.
Kurzschlussabschaltung		Ja (DC 5 V dauerkurzschlussfest)
Ausgangsspannungsüberwachung		Ja (DC 5 V wird auf Über- und Unterspannung überwacht)
Galvanische Trennung		nein
Versorgungsspannung $24V_{ext}$, M_{ext}		DC 24 V (19,2 V .. 30 V einschl. 5% Restwelligkeit)
Eingangsstrom		max. 3 A, bei max. Belastung der Ausgänge
Galvanische Trennung (zur internen Elektronik)	X1, X2 SERCOS III	Ja
	X3 Ethernet	Ja
	X4 USB Host	Nein
	X5 USB Device	Nein
	X6 CAN	Ja
	X7 RS422	Ja
	X7 RS232	Ja
	X9 I/Os 24V	Ja
	X10 Inputs 5V	Ja
	X11 Outputs 5V	Ja

Eingänge 24 V	
Schaltpegel	H-Pegel +15 V bis +30 V L-Pegel -30 V bis +5 V
Eingangsstrom	min. H-Pegel (+15 V), I ≥ 2,3 mA / 3,5 mA* max. L-Pegel (+ 5V), I ≤ 0,9 mA / 1,4 mA* typisch (+24 V), I = 4,5 mA / 6,1 mA* * für Kombi-I/O
Signalverzögerung	<300 µs (Hardware)
Eingänge 5 V (differentiell)	
Schaltpegel	RS422
Eingangsstrom	max. 50 mA (Uin = 5 V)
Signalverzögerung	<200 ns (Hardware)
gefilterte Versorgungsspannung 24VF zur Versorgung von Gebern	
Ausgangsstrom max.	250 mA

Ausgänge 24 V	
Schaltpegel	H-Pegel ≥ 24Vext - 0,5 V L-Pegel ≤ 1 V
Ausgangsstrom je Ausgang max.	500 mA überstrom- und kurzschlussfest
Signalverzögerung	<300 µs (Hardware)

Ausgänge 5 V (differentiell)	
Schaltpegel	H-Pegel ≥ 4,2 V (I_L = -20mA) L-Pegel ≤ 0,5 V (I_L = 20mA)
Ausgangsstrom je Ausgang max.	50 mA kurzschlussfest
Signalverzögerung	<400 ns (Hardware)

Schnittstellen		
Ethernet	RJ 45	Programmier-, Diagnose und Bediengeräteschnittstelle
sercos III (nur XCx 800)	RJ 45	sercos III Antriebsschnittstelle (Ethernet)
USB Host	Standard A	USB-Schnittstelle (USB-Speicherstick)
USB Device	Standard B	USB-Schnittstelle (PC)
CAN	10-pol. Steckblockklemme	CANopen Feldbusschnittstelle
RS 232	10-pol. Steckblockklemme	für stationären Anschluss serieller Geräte
RS 422	10-pol. Steckblockklemme	serielle Bediengeräteschnittstelle
Digitale I/Os	10-pol. Steckblockklemme	Lokale Ein-/Ausgänge
XRIO	10-pol. Steckblockklemme	XRIO Busschnittstelle

Sonstige technische Daten		
Prozessor		i.MX31L ARM
Speicher- ausstattung	DDR-SDRAM	64 MB
	SRAM (gepuffert)	1 MB
	FLASH (intern)	2 MB
	Compact FLASH	>= 512 MB
Real-Time Clock		Batteriegepuffert mit Kalender und Schaltjahr, Auflösung: 1s
Pufferung		3 Monate über wiederaufladbare Batterie (Ladezeit mini. 4h)
Display (nur XCx800)		graphisch, 128 x 32 Pixel

CNC/SPS-Eigenschaften		
SPS-Bearbeitungs-zeiten je 1000 Anweisungen	Bit	0,237 ms
	Byte / Word / DWord	0,104 ms
	Integer (Add / Mul)	0,103 ms
	Real (Add)	0,677 ms
SPS-Signallaufzeit (Input to Output)		< 2 ms (bei Taskperiode = 1 ms)
Funktionsbausteine		Firmwarefunktionen und Funktionsbausteine in beliebiger Anzahl
Anzahl NC-Achsen / Teilsysteme (XCN)		32 / 16
CNC-Interpolationstakt (XCN) , ab		1 ms
Blockzykluszeit, ab		1 ms
Betriebs-system	Steuerung	VxWorks, Multitask-Betriebssystem (zeit- und prioritätsgesteuert)
	SPS-Runtime	ProConOS
Projektierung		MULTIPROG nach IEC 61131-3
Anzahl der Anwendertasks		18
Taskzykluszeiten		programmierbar ≥ 1 ms (ganzzahlig)
Echtzeit-Speicher	Betriebssystem (Daten / Programme)	32768 kB
SPS-Speicher	Programme	4096 kB
	Merker remanent	256 kB
	Merker nicht remanent	2048 kB
Speicherverwaltung		dynamisch
Zeiten und Zähler		beliebig viele programmierbar von 1 ms ... 290 h (Anzahl nur durch Speicherauslastung begrenzt)

Abmessungen / Gewicht	
Maße (B x H x T)	71 mm x 200 mm x 150 mm
Teilungsbreite	2
Gewicht	1000 g

Zusätzlich gelten die Angaben im Kapitel "Technische Daten aller Module", Seite 134.

4.4 Bestellangaben XCx 400 / 800

Bestellbezeichnung	Ausstattung	Artikelnummer
XCS 400 C	SPS-Steuereinheit	R4.501.0200.0
XCN 400 C2	CNC-Steuereinheit (max. 2 CNC-Achsen)	R4.501.0270.0
XCN 400 CE	CNC-Steuereinheit (max. 4 CNC-Achsen)	R4.501.0230.0
XCN 400 C6	CNC-Steuereinheit (max. 6 CNC-Achsen)	R4.501.0220.0
XCN 400 C8	CNC-Steuereinheit (max. 8 CNC-Achsen)	R4.501.0280.0
XCN 400 C16	CNC-Steuereinheit (max. 16 CNC-Achsen)	R4.501.0290.0
XCS 800 CS	SPS-Steuereinheit	R4.501.0210.0
XCN 800 CS2	CNC-Steuereinheit (max. 2 CNC-Achsen plus 2 sercos III Achsen)	R4.501.0300.0
XCN 800 CES	CNC-Steuereinheit (max. 4 CNC-Achsen plus 4 sercos III Achsen)	R4.501.0250.0
XCN 800 CS6	CNC-Steuereinheit (max. 6 CNC-Achsen plus 6 sercos III Achsen)	R4.501.0310.0
XCN 800 CS8	CNC-Steuereinheit (max. 8 CNC-Achsen plus 8 sercos III Achsen)	R4.501.0240.0
XCN 800 CS12	CNC-Steuereinheit (max. 12 CNC-Achsen plus 12 sercos III Achsen)	R4.501.0360.0
XCN 800 CS16	CNC-Steuereinheit (max. 16 CNC-Achsen plus 16 sercos III Achsen)	R4.501.0260.0
XCN 800 CS24	CNC-Steuereinheit (max. 24 CNC-Achsen plus 24 sercos III Achsen)	R4.501.0320.0
XCN 800 CS32	CNC-Steuereinheit (max. 32 CNC-Achsen plus 32 sercos III Achsen)	R4.501.0330.0

5 Erweiterungsmodule XCx und Promodul-U

Für die Steuereinheiten XCx 400 / XCx 800 stehen eine Vielzahl an Baugruppenträgern, Netzgeräten und Erweiterungsmodulen zur Verfügung. Diese Module werden in einer eigenen Betriebsanleitung beschrieben (→ Seite 10).

Modul	Artikel-Nr.	Bemerkung
Baugruppenträger		
XBT 2	R4.507.0140.0	Basis-/Erweiterungsbaugruppenträger, 2 Steckplätze
XBT 4	R4.507.0150.0	Basis-/Erweiterungsbaugruppenträger, 4 Steckplätze
XBT 8	R4.507.0160.0	Basis-/Erweiterungsbaugruppenträger, 8 Steckplätze
XBT 12	R4.507.0170.0	Basis-/Erweiterungsbaugruppenträger, 12 Steckpl.
XBT 16	R4.507.0180.0	Basis-/Erweiterungsbaugruppenträger, 16 Steckpl.
UBT 4	R4.311.0010.0	Basis-/Erweiterungsbaugruppenträger, 4 Steckplätze
UBT 8	R4.311.0020.0	Basis-/Erweiterungsbaugruppenträger, 8 Steckplätze
UBT 12	R4.311.0030.0	Basis-/Erweiterungsbaugruppenträger, 12 Steckpl.
UBT 16	R4.311.0040.0	Basis-/Erweiterungsbaugruppenträger, 16 Steckpl.
UBT 20	R4.311.0050.0	Basisbaugruppenträger, 20 Steckplätze
Koppelmodule		
UKZ	R4.318.0030.0	Koppelmodul für Basisbaugruppenträger
UKE	R4.318.0040.B	Koppelmodul für Erweiterungsbaugruppenträger
Netzgeräte		
UNG 230A x*	R4.312.0030.F	Netzgerät 230 V, Breite 2 Einheiten
UNG 115A x*	R4.312.0040.F	Netzgerät 115 V, Breite 2 Einheiten
UNG 24 x*	R4.312.0020.B	Netzgerät 24 V, Breite 1 Einheit
Digitale E/A-Module		
UBE 32 0,1I	R4.314.0100.E	32 Eingänge, 4 Interrupts, 0,1 ms Eingangsverz.
UBE 32 1D	R4.314.0120.E	32 Eingänge, 1 ms Eingangsverzögerung
UBE 32 10D	R4.314.0090.E	32 Eingänge, 10 ms Eingangsverzögerung
UBA 32/2A	R4.314.0080.D	32 Halbleiterausgänge DC 24V / 2A
UBK 16E 1D/16A	R4.314.0130.E	16 Eingänge, 1 ms Eingangsverz. / 16 Ausgänge
UBK 16E 10D/16A	R4.314.0110.E	16 Eingänge, 10 ms Eingangsverz. / 16 Ausgänge
XBE 32 0,1I	R4.314.0170.0	32 Eingänge, 4 Interrupts, 0,1 ms Eingangsverz.
XBE 32 1D	R4.314.0140.0	32 Eingänge, 1 ms Eingangsverzögerung
XBE 32 10D	R4.314.0180.0	32 Eingänge, 10 ms Eingangsverzögerung
XBA 32/1A	R4.314.0150.0	32 Halbleiterausgänge DC 24V / 1A
XBK 16E 1D/16A/1A	R4.314.0160.0	16 Eingänge, 1 ms Eingangsverz. / 16 Ausgänge /1A
XBK 16E 10D/16A/1A	R4.314.0190.0	16 Eingänge, 10 ms Eingangsverz. / 16 Ausgänge /1A
Zählermodule		
UZB 2VR	R4.315.0010.B	2 Zähler, 24 V Eingangsspannung
UZB 2VR/5V	R4.315.0040.B	2 Zähler, 5 V Eingangsspannung

Modul	Artikel-Nr.	Bemerkung
Multifunktionsmodule		
XSF 05	R4.315..0340.0	Funktionsmodul UNIVERSAL, 5V-Pegel
XSF 24	R4.315..0350.0	Funktionsmodul UNIVERSAL, 24V-Pegel
XSL 05	R4.315..0360.0	Funktionsmodul LASER, 5V-Pegel
XSL 24	R4.315..0370.0	Funktionsmodul LASER, 24V-Pegel
Analog- und Temperaturmodule		
USA 8/1	R4.315.0090.F	Analogprozessor, 8 Slots für USA-Module
USA E1/1	R4.315.0100.0	1 Spannungseingang
USA E1/2.1	R4.315.0120.0	1 Stromeingang
USA E1/6	R4.315.0140.0	1 Widerstandstemperaturmessung Pt100
USA E1/7	R4.315.0150.0	1 Thermoelementeingang Fe-CuNi
USA A1/1	R4.315.0110.B	1 Spannungsausgang
USA A1/2	R4.315.0130.0	1 Stromausgang
Positioniermodule		
XSP 200	R4.508.0100.0	sercos 2 Master, 1 Ring, 8 Achsen
XSP 400	R4.508.0200.0	sercos 2 Master, 2 Ringe, 16 Achsen
USP 200S	R4.315.0300.0	sercos 2 Master, 1 Ring, 8 Achsen
USP 400S	R4.315.0330.0	sercos 2 Master, 2 Ringe, 16 Achsen
USP 2I	R4.315.0020.0	Positionierprozessor, 2 Achsen, Inkremental-Encoder
USP 2A	R4.315.0030.0	Positionierprozessor, 2 Achsen, Absolut-Encoder SSI
UPI 2 DIA	R4.318.0180.B	Positionierinterface, 2 Achsen
UPI 3 DIA	R4.318.0160.B	Positionierinterface, 3 Achsen
Kommunikationsmodule		
USK DIM	R4.318.0170.0	Interbus-S-Master
USK DPM	R4.318.0370.0	PROFIBUS-DP-Master
USK DPS	R4.318.0360.0	PROFIBUS-DP-Slave
Zubehör		
UBT LA	R4.318.0120.0	Leerort-Abdeckungen für UBT
UKK 24	R4.318.0020.0	Kabel UKZ ↔ UKE, ohne Spannungsversorgung
UKK 24V	R4.318.0060.0	Kabel UKZ ↔ UKE, mit Spannungsversorgung
UNB 115/230	R4.318.0050.0	Pufferbatterie für UNG 230A/115A
UNB 24	R4.318.0130.0	Pufferbatterie für UNG 24

Tabelle 18: Liste der verfügbaren

x* = Verwendung nur auf Erweiterungsbaugruppenträgern

6 Inbetriebnahme

Die in diesem Kapitel beschriebene Inbetriebnahme der XCx kann ohne tiefgreifendes Wissen durchgeführt werden. Die Inbetriebnahmeschritte müssen genau befolgt und die Rahmenbedingungen (wie z. B. die I/O-Konfiguration) eingehalten werden.

Hinweis

Die in den folgenden Kapiteln gezeigten Screenshots zu Software-Installation und Inbetriebnahme sind beispielhaft. Versionsnummern von Software oder Gerätebezeichnungen können vom aktuellen Stand abweichen.

6.1 Installation von MULTIPROG, OPC-Server und Add-Ons

Wichtig!

Die gesamte Programmiersoftware besteht aus den Software-Komponenten MULTIPROG, dem OPC-Server, Add-Ons für MULTIPROG und dem Schleicher-Dialog.

Alle Software-Komponenten müssen vor der weiteren Inbetriebnahme einzeln, in dieser Reihenfolge nacheinander installiert werden.

Die Software ist als Service CD-ROM auf Anfrage verfügbar:

Tabelle 19: Inhalt der CD-ROM

Name	Inhalt
MULTIPROG	• Programmiersoftware MULTIPROG • OPC-Server
Service Pack	• Steuerungssoftware für alle Schleicher-Steuerungen • Add-Ons • Schleicher-Dialog • Weitere Hilfsmittel wie Dokumentation und Service-Informationen

6.2 Systemvoraussetzungen

Für Installation und Betrieb der Software sind folgende Systemvoraussetzungen einzuhalten:

Tabelle 20: Systemvoraussetzungen

Windows-PC	Pentium, 1 GHz
Arbeitsspeicher	256 MB
Festplatte	250 MB
Monitor	1024 x 768 (True Color)
Schnittstellen	Ethernet (RS232 für Diagnose)
CD-ROM Laufwerk	
PC-Betriebssystem	Windows 2000(SP2), XP, Vista

6.2.1 MULTIPROG installieren

Die CD MULTIPROG in das Laufwerk des PC einlegen. Über die AutoRun-Funktion der CD wird der Internet Explorer gestartet. Dann *MULTIPROG* (hier Version 4.8) auswählen und die Installation starten (→ Bild 7).

Bild 7: Installation von MULTIPROG

schleicher
control systems
MULTIPROG® 4.8
Willkommen zur Installations-CD für das SPS-Programmiersystem MULTIPROG® und den ProConOS® OPC-Server. MULTIPROG® und der ProConOS® OPC-Server können für die folgenden SPS-/CNC-Steuerungen von Schleicher verwendet werden:
XCN 3xx, XCS 3xx
XCN 4xx, XCS 4xx
XCN 5xx, XCS 5xx
XCN 7xx, XCS 7xx
XCN 8xx, XCS 8xx
XCA 11xx
ProNumeric, ProSyCon
Microline MCS 20-20, MCS 20-21
Software
Wählen Sie einen der folgenden Links, um das Setup-Programm für die entsprechende Software aufzurufen.
Aktuelle Versionen:
MULTIPROG® 4.8, Build 332 mit ProVisIT 2.3, Build 758
ProConOS® OPC-Server 2.1, Build 187

Hinweis

Ist eine Version von MULTIPROG kleiner als 4 bereits installiert, darf die installierte Version nicht überschrieben werden, wenn weiter mit den alten Projekten gearbeitet werden soll.

MULTIPROG muss dann in einem neuen Pfad installiert werden. Alle anderen Installations-Einstellungen können unverändert bleiben.

Am Ende der Installation erfolgt eine Aufforderung zum Neustart des Computers. Falls Sie anschließend auch den ProConOS OPC-Server installieren wollen, ist ein Neustart des Computers jetzt noch nicht erforderlich.

6.2.2 OPC-Server installieren

Zur Installation des OPC-Servers im Internet Explorer den *ProConOS OPC-Server* auswählen und die Installation starten (→ Bild 8).

Bild 8: Installation des OPC-Servers

schleicher
control systems
MULTIPROG® 4.8
Willkommen zur Installations-CD für das SPS-Programmiersystem MULTIPROG® und den ProConOS® OPC-Server.
MULTIPROG® und der ProConOS® OPC-Server können für die folgenden SPS-/CNC-Steuerungen von Schleicher verwendet werden:
XCN 3xx, XCS 3xx
XCN 4xx, XCS 4xx
XCN 5xx, XCS 5xx
XCN 7xx, XCS 7xx
XCN 8xx, XCS 8xx
XCA 11xx
ProNumeric, ProSyCon
Microline MCS 20-20, MCS 20-21
Software
Wählen Sie einen der folgenden Links, um das Setup-Programm für die entsprechende Software aufzurufen.
Aktuelle Versionen:
MULTIPROG® 4.8, Build 332
mit ProVisIT 2.3, Build 758
ProConOS® OPC-Server 2.1, Build 187

Der OPC-Server soll im MULTIPROG-Pfad installiert werden. Alle anderen Installations-Einstellungen können unverändert bleiben. Nach der Installation ist ein Neustart des PCs erforderlich.

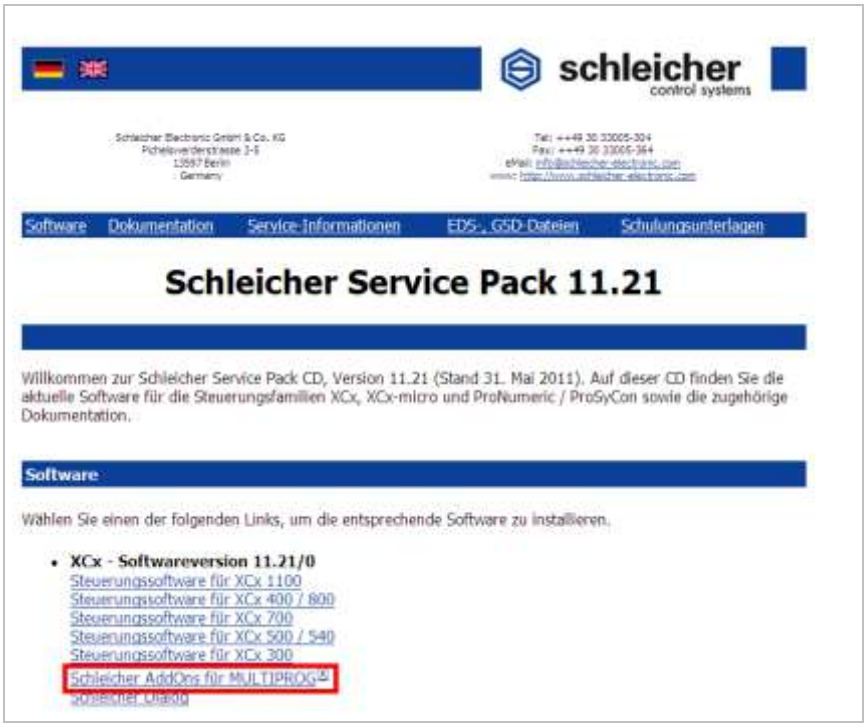
DenOPC-Server Einrichten siehe -> Vorbereitung des OPC-Servers auf Seite 95.

6.2.3 Add-Ons installieren

Als nächster Schritt müssen die Add-Ons für MULTIPROG installiert werden.

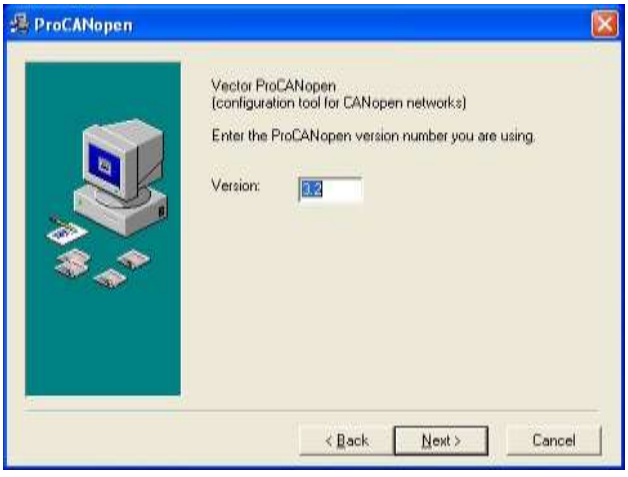
Legen Sie dazu die CD *Service Pack* ein. Über die AutoRun-Funktion der CD wird der Internet Explorer gestartet. Dann unter der Rubrik für die vorhandene Steuerung *Add-Ons für MULTIPROG* auswählen und die Installation starten (→ Bild 9).

Bild 9: Installation der Add-Ons


schleicher
control systems
Software
Dokumentation
Service-Informationen
EDS-, GSD-Dateien
Schulungsunterlagen
Schleicher Service Pack 11.21
Willkommen zur Schleicher Service Pack CD, Version 11.21 (Stand 31. Mai 2011). Auf dieser CD finden Sie die aktuelle Software für die Steuerungsfamilien XCx, XCx-micro und ProNumeric / ProSyCon sowie die zugehörige Dokumentation.
Software
Wählen Sie einen der folgenden Links, um die entsprechende Software zu installieren.
XCx - Softwareversion 11.21/0
Steuerungssoftware für XCx 1100
Steuerungssoftware für XCx 400 / 800
Steuerungssoftware für XCx 700
Steuerungssoftware für XCx 500 / 540
Steuerungssoftware für XCx 300
Schleicher AddOns für MULTIPROG

Im Lauf der Installation muss die vom Nutzer aktuell verwendete ProCANopen-Version (hier 3.2) eingetragen werden (→ Bild 10). ProCANopen wird später zur Inbetriebnahme des CANopen-Netzes benötigt. Wird kein ProCANopen eingesetzt, können die Voreinstellungen übernommen werden.

Bild 10: AddOn-Installation, Eingabe der ProCANopen-Version



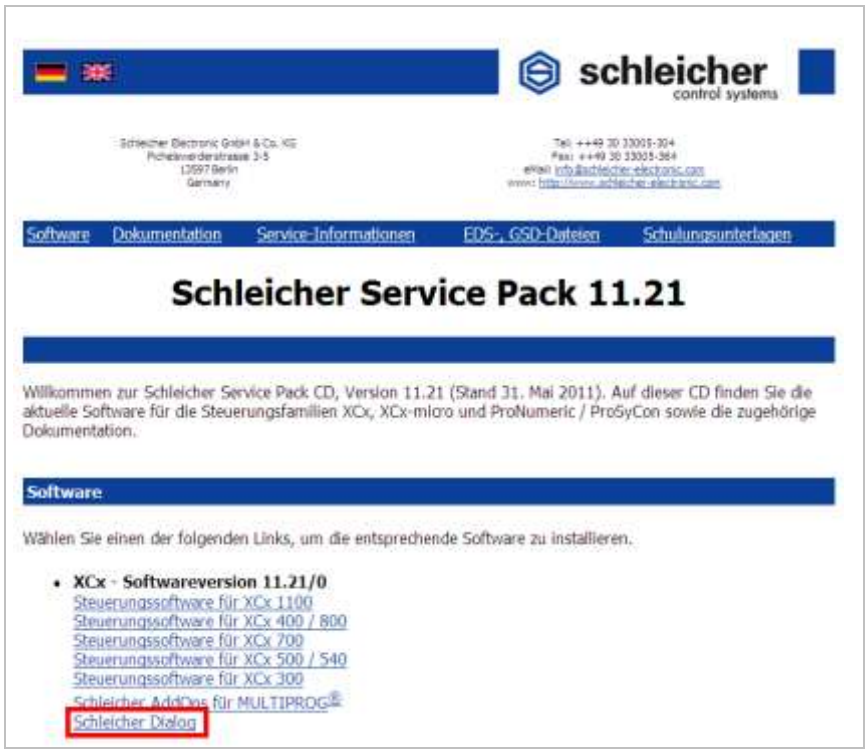
Ein Neustart des PCs nach der Installation ist nicht erforderlich.

6.2.4 Schleicher Dialog installieren

Anschließend wird die Bedienoberfläche Schleicher Dialog installiert. Dazu im Internet Explorer unter der Rubrik für die vorhandene Steuerung *Dialog* auswählen und starten (→ Bild 11).

Der Schleicher Dialog wird beschrieben im Kapitel Bedienung der XCx mit dem Schleicher Dialog auf Seite 94.

Bild 11: Installation des Schleicher-Dialogs


schleicher
control systems
Software
Dokumentation
Service-Informationen
EDS-, GSD-Dateien
Schulungsunterlagen
Schleicher Service Pack 11.21
Willkommen zur Schleicher Service Pack CD, Version 11.21 (Stand 31. Mai 2011). Auf dieser CD finden Sie die aktuelle Software für die Steuerungsfamilien XCx, XCx-micro und ProNumeric / ProSyCon sowie die zugehörige Dokumentation.
Software
Wählen Sie einen der folgenden Links, um die entsprechende Software zu installieren.
XCx - Softwareversion 11.21/0
Steuerungssoftware für XCx 1100
Steuerungssoftware für XCx 400 / 800
Steuerungssoftware für XCx 700
Steuerungssoftware für XCx 500 / 540
Steuerungssoftware für XCx 300
Schleicher AddOns für MULTIPROG®
Schleicher Dialog

6.3 Inbetriebnahme der Netzwerk-Schnittstelle

Wichtig!

Die netzwerkspezifischen Besonderheiten und die Vorgehensweise sind zuerst mit dem Netzwerkadministrator für das jeweilige Hausnetz zu klären.

Alle in den folgenden Installationshinweisen angeführten bzw. vorgegebenen Bezeichner und Adressen sind beispielhaft und müssen gegebenenfalls an Ihre lokalen Verhältnisse angepasst werden.

Alle Beispiele dieser Anleitung sind mit Windows XP erstellt. Die Vorgehensweise in anderen Betriebssystemen kann von der hier beschriebenen abweichen.

Die hier gegebenen Hinweise sind in jedem Falle unverbindlich!

6.3.1 Kommunikationsschema

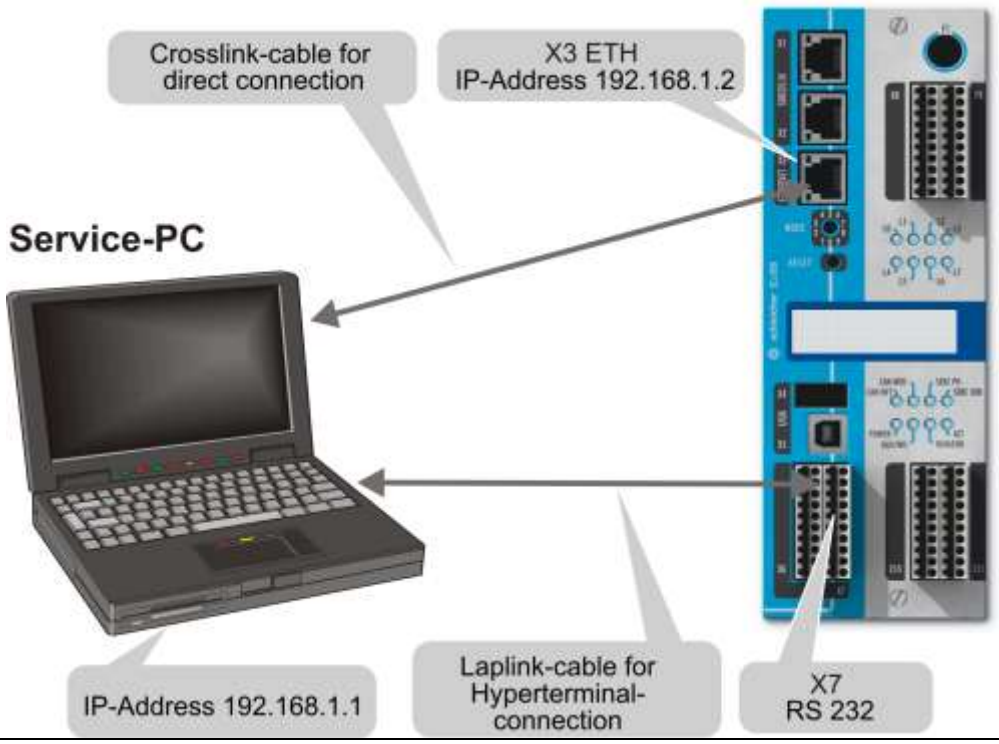


Bild 12: Kommunikationsschema

6.4 Inbetriebnahme der Ethernet-Schnittstelle

Die Inbetriebnahme, Konfiguration und Programmierung der XCx erfolgt über die Ethernet- Schnittstelle. .

Voraussetzung ist ein PC mit Netzwerkkarte für Ethernet und ein Cross-Link oder Link-Kabel für die Direktverbindung zur XCx.

6.4.1 Einstellen der IP-Adresse des Service-PC

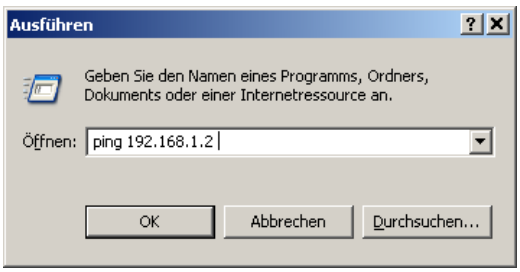
- Die IP-Adresse des Service-PC wird unter Netzwerkverbindungen im Bereich Systemsteuerung eingestellt.
- Zum Einstellen ist gegebenenfalls eine Netzwerkverbindung erforderlich.
  Dann TCP/IP auswählen und Eigenschaften anklicken.
  IP-Adresse eingeben: 192.168.1.1
  Subnet Mask eingeben: 255.255.255.0
- Überprüfen der IP-Adresse des Service-PCs, hierzu ist in Windows das Fenster *Eingabeaufforderung* zu öffnen. Mit Aufruf von *ipconfig* wird die aktive Einstellung angezeigt.
- Als IP-Adresse sollte jetzt **192.168.1.1** stehen.
  Als Subnet Mask sollte jetzt **255.255.255.0** stehen.

Zum Beenden das Fenster schliessen.

6.4.2 Ethernet-Verbindung herstellen und prüfen

- Den Ethernet-Anschluss des PC mit der Ethernet-Schnittstelle "Ethernet X3" der XCx über ein Cross-Link-Kabel verbinden.
- Auf dem PC *Start/Ausführen* wählen.
- In das Eingabefeld <Öffnen:> ping 192.168.1.2 eintragen (diese IP-Adresse ist bei Auslieferung der XCx voreingestellt).

Bild 13: Ethernet Verbindung prüfen



- Mit O.K. starten,
  ping wird ausgeführt für 192.168.1.2 mit 32 Bytes Daten:

  Anzeige bei korrekter Verbindung:
  Antwort von 192.168.1.2: Bytes=32 Time<1ms TTL=63
  Antwort von 192.168.1.2: Bytes=32 Time<1ms TTL=63
  Antwort von 192.168.1.2: Bytes=32 Time<1ms TTL=63
  Antwort von 192.168.1.2: Bytes=32 Time=1ms TTL=63

  Anzeige bei nicht korrekter Verbindung:
  Zeitüberschreitung der Anforderung.
  Zeitüberschreitung der Anforderung.
  Es sind die IP-Adresse und die Kabelanschlüsse zu überprüfen.

6.5 Ändern der IP-Adresse der XCx

Das Ändern der IP-Adresse ist nicht notwendig, wenn die XCx wie vorher beschrieben über ein Crosslink-Kabel direkt am PC betrieben wird.

- Vorzugsweise

Wichtig!

Vor dem Ändern der IP-Adressen die Vorgehensweise zuerst mit dem Netzwerkadministrator für das jeweilige Hausnetz klären.

Die hier gegebenen Hinweise sind in jedem Falle unverbindlich.

Zum Ändern der IP-Adresse ist ein Menü im Schleicher Dialog vorhanden. Die Inbetriebnahme vom Schleicher Dialog siehe Seite 99.

Bild 14:
IP-Adresse ändern im Schleicher Dialog

Nach der Eingabe aller IP-Daten, der Benutzername und das Kennwort dürfen nicht gleich sein, den Dialog verlassen und die Übernahme der Änderung bestätigen.

Die IP-Einstellung ist damit in der XCx in dem Verzeichnis \OS in der Datei BootLine.ini gespeichert.

Zur Übernahme der Adressänderung muss die XCx in der Mode-Schalterstellung 1 oder 0 neu gestartet werden.

Wichtig!

Die bisherige IP-Einstellung der XCx bleibt solange aktiv bis die Änderung der IP-Einstellung mit der Mode-Schalterstellung 1 bzw. 0 und Neustart ausgeführt wird.

Anschließend den Service-PC (Netzwerkverbindung), den Schleicher Dialog, den OPC-Server und das SPS-Projekt in MULTIPROG auf die im Dialog eingestellte Adresse anpassen.

- Alternativ

Es besteht die Möglichkeit die IP-Einstellung über für den FTP(File Transfer Protokoll)-Zugriff (z.B.Windows® Internet Explorer 6) zu ändern

Wichtig!

Für den Zugriff per FTP muß die TCP/IP Verbindung zur XCx aktiv sein. (Siehe Überprüfen der IP-Adresse mit Ping: Abschnitt 6.4.2.)

Mit Eingabe der bekannten IP-Einstellung wird die Verbindung zur XCx hergestellt.

Von dem Compact Flash wird aus dem Verzeichniss \OS die Datei BootLine.ini auf die Festplatte kopiert.

In die Datei BootLine.ini werden die gewünschten IP-Einstellungen eingetragen und anschliessend wird die Datei auf das Compact Flash zurück geschrieben. Gegebenenfalls muss das Nur-Lesen-Attribut über die Verbindung mit der RS232-Schnittstelle mit den Befehl attrib "bootline.txt","-R" eingeben und ausführen.

Tabelle 21: Bezeichnungen der Netzwerkeinstellungen

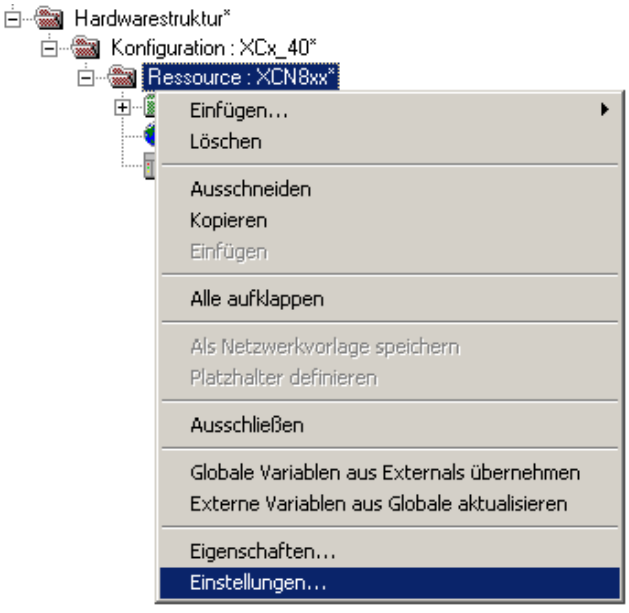
Schleicher Dialog	BootLine.ini	Netzwerk-Einstellung
IP-Adresse der Steuerung	InetOnEthernet	192.168.1.2
Subnetzmaske	InetMask	255.255.255.0
Benutzername	UserName	schleicher
Passwort	Password	berlin
Name der Steuerung	TargetName	XCx

6.5.1 Kommunikation zur Programmierung mit MULTIPROG

Auf dem externen PC kann jetzt MULTIPROG gestartet werden. Wählen Sie *Neues Projekt / XCN8xx* (oder öffnen Sie ein bereits bestehendes Projekt für die vorhandene Steuerung). Eine ausführliche Beschreibung der Vorgehensweise finden Sie im Kapitel " Erste Schritte mit MULTIPROG" auf Seite 41.

Im SPS-Projekt klicken Sie mit der rechten Maustaste auf den Eintrag *Ressource: XCN8xx* und wählen im Kontextmenü *Einstellungen* (→ Bild 15).

Bild 15: Aufruf der Ressource-Einstellungen in MULTIPROG

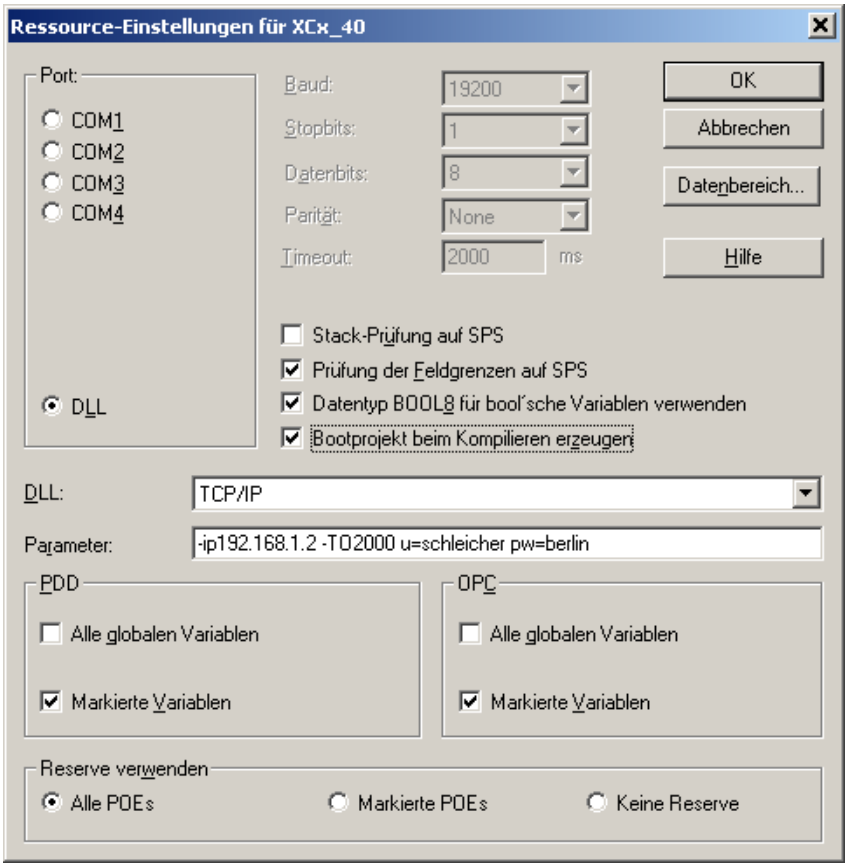


Im Dialogfenster *Ressource-Einstellungen für XCx_40* wird die IP-Adresse angegeben (→ Bild 16).

Beispiel (wenn alle Default-Einstellungen übernommen wurden):

```
-ip192.168.1.2 -TO2000 u=schleicher pw=berlin
```

Bild 16:
Eingeben der IP-Adresse
in MULTIPROG



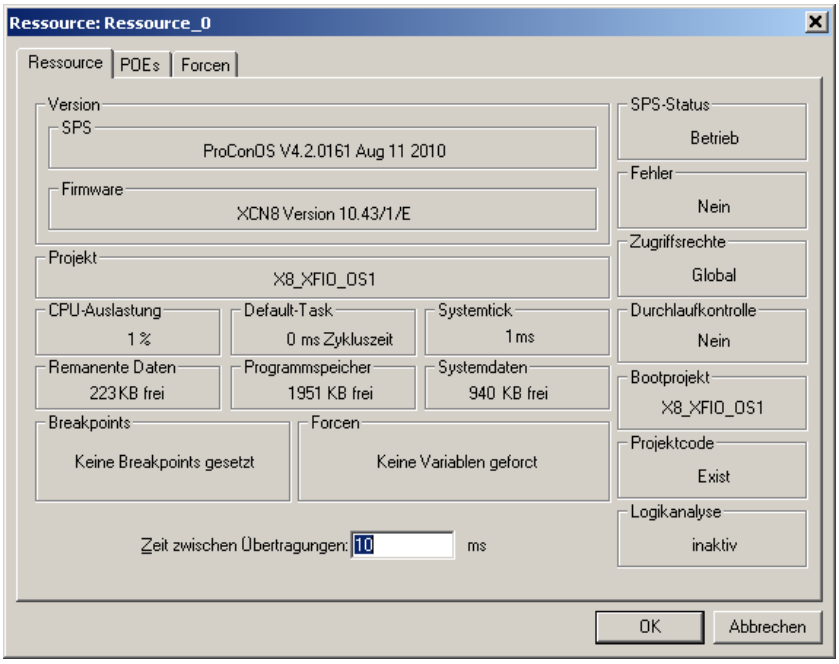
Nach Bestätigen mit *OK* wird die Verbindung zur XCx 800 mit *Online / Projektkontrolle* und anschließend im Fenster *Ressource* mit dem Button *Info* aufgerufen (→ Bild 17).

Bild 17:
Aufruf der Verbindung zur XCx 400 / XCx 800
in MULTIPROG

Mit erfolgreicher Verbindung wird das Fenster ´Ressource´ angezeigt (→ Bild 18). Bei fehlerhafter Verbindung wird *Timeout* gemeldet. In diesem Falle sind alle Einstellungen nochmals zu überprüfen oder der Netzwerkadministrator zu Rate zu ziehen.

Eventuell sind weitere Einstellungen bei Gateways zur Subnetzvermittlung etc. zu beachten.

Bild 18: Ressource-Fenster bei Online-Verbindung zur XCx 800



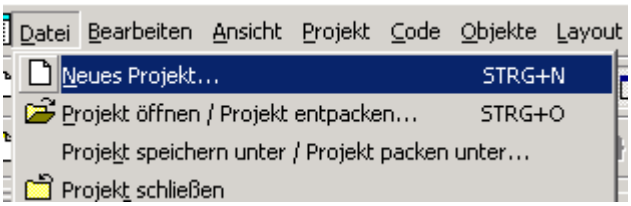
6.6 Erste Schritte mit MULTIPROG

Für weitere Information zur Inbetriebnahme von MULTIPROG siehe auch die BA "Getting Started MULTIPROG" (→ Seite 10).

6.6.1 MULTIPROG starten, neues Projekt öffnen und speichern

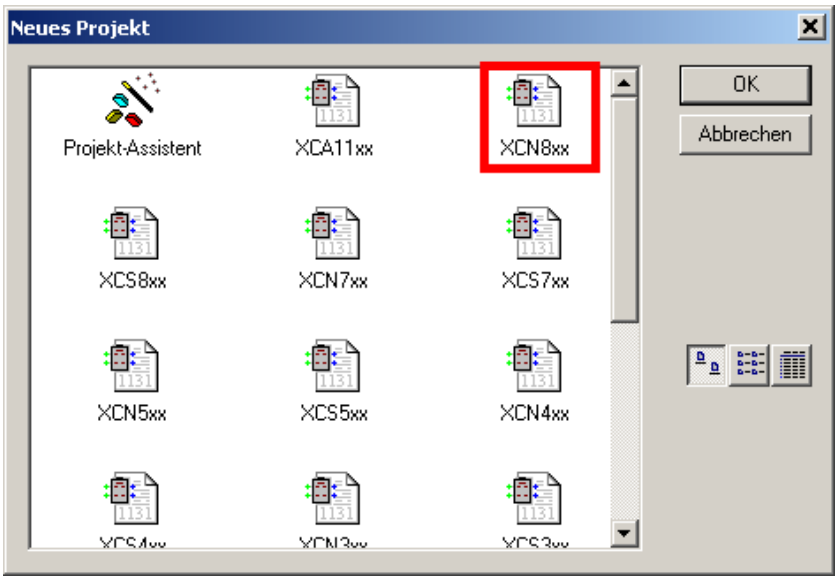
MULTIPROG starten, *Datei / Neues Projekt* wählen (→ Bild 19).

Bild 19: Neues Projekt in MULTIPROG öffnen



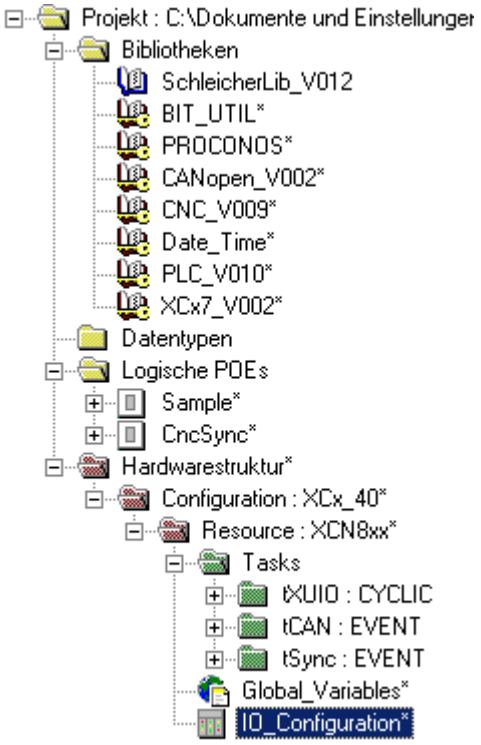
Ein Projekt für den vorhandenen Steuerungstyp (hier XCN 8xx) auswählen, mit *OK* ausführen (→ Bild 20).

Bild 20: Steuerungstyp für neues Projekt auswählen



Ist das Projekt erfolgreich geöffnet, wird im Projektfenster der Projektbaum dargestellt (→ Bild 21). Es sind bereits Logische POEs enthalten, die voll funktionsfähig sind und für eine einfache Schnellinbetriebnahme ausreichen.

Bild 21: Beispiel für Projektbaum in MULTIPROG



Unter *Datei / Projekt speichern unter* das Projekt mit einem neuen Namen (hier START) speichern (→ Bild 22 und Bild 23).

Bild 22: Speicherdialog aufrufen

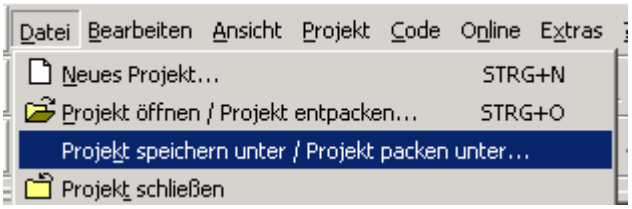
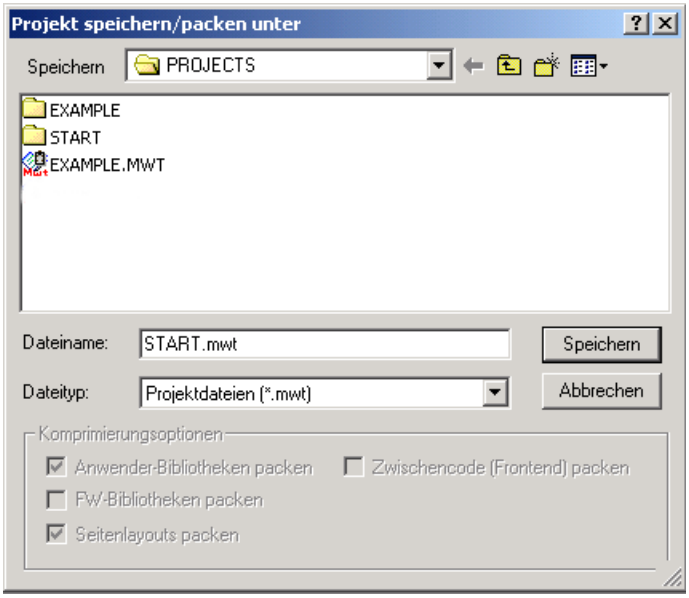


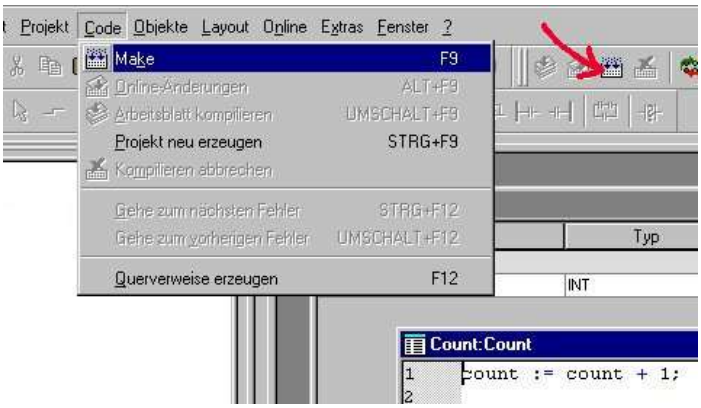
Bild 23: Dialogfenster "Projekt speichern"



6.6.2 Ein Projekt kompilieren und zur XCx senden

Um das Projekt zu kompilieren, *Code / Make* wählen (→ Bild 24) (alternativ <F9> oder Button *Make,* siehe Pfeil).

Bild 24:
Projekt kompilieren



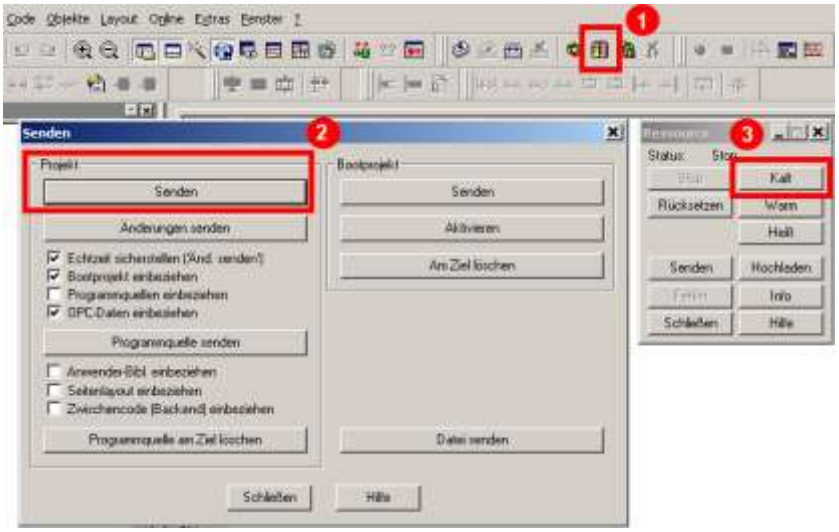
Im Meldungsfenster unten wird der Kompiler-Lauf protokolliert. Hier werden auch Fehlermeldungen, Warnungen und weitere Infos angezeigt. Ein Anwählen der Auswahlbox "Fehler", "Warnungen" usw. zeigt die Meldung detaillierter an.

Werden Fehler angezeigt, kann mit Doppelklick auf der Fehlerzeile direkt in die Zeile des SPS-Programms, die den Fehler verursachte, verzweigt werden.

Anschließend übertragen Sie das Projekt über die Ethernet-Verbindung auf die Steuerung (→ Bild 25):

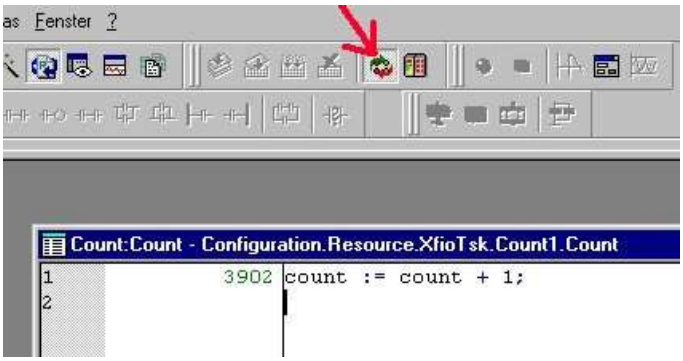
1. Auf den Button *Projekt-Kontrolldialog* . Im sich daraufhin öffnenden Kontrolldialog *XCx* auf den Button *Senden* klicken.
2. Im Bereich *Projekt* wieder *Senden* wählen, damit wird das auf der XCx vorhandene Projekt überschrieben.
3. Mit dem Button *Kalt* (Kaltstart) im Kontrolldialog *Ressource* wird das Programm auf der XCx gestartet. Die LED *PLC RUN* auf der Steuerung leuchtet dauernd grün.

Bild 25:
Projekt auf die Steuerung übertragen



Mit dem Button *Debug ein/aus* (siehe Pfeil) kann auf dem Arbeitsblatt der Inhalt der Variablen online angezeigt werden (→ Bild 26).

Bild 26:
Online-Anzeige der Variablen



6.7 Einfügen des Koppelspeichers

Um den vollen Zugriff auf alle vordefinierten Variablen zu erlangen, ist es an dieser Stelle der Inbetriebnahme angebracht, die Koppelspeicherstruktur in das Projekt einzufügen.

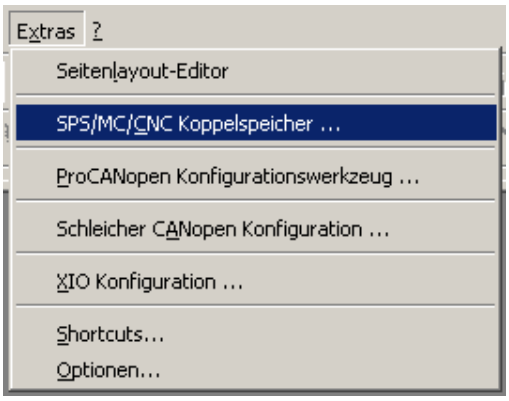
Der Koppelspeicher besitzt eine umfangreiche Datenstruktur, die zur Kommunikation zwischen SPS, CNC und Visualisierungssystemen verwendet wird. Außerdem kann über den Koppelspeicher auf Datenbereiche wie z.B. Version Number, Error Page und Log Book zugegriffen werden.

Eine Einführung zum Thema Koppelspeicher finden Sie auf Seite 106. Eine genaue Beschreibung sämtlicher Variablen des Koppelspeichers ist als Online-Hilfe der Software "Schleicher Dialog" verfügbar. Díese kann auch über das Hilfe-Menü von MULTIPROG aufgerufen werden.

Um auf Variablen des Koppelspeichers zugreifen zu können, müssen die entsprechenden Datentypen und Variablen in das SPS-Projekt eingefügt werden. Diese sind in den Projekt-Templates von MULTIPROG noch nicht enthalten. Sie müssen vom Anwender selbst mit Hilfe des Koppelspeicher-Add-Ons für MULTIPROG in das SPS-Projekt eingefügt werden. Damit soll sichergestellt werden, dass der Anwender mit der für die Steuerungssoftware passenden Koppelspeicher-Variante arbeitet.

Das Einfügen der Koppelspeicher-Datentypen und -Variablen in ein SPS-Projekt erfolgt über das Menü *Extras / SPS/MC/CNC Koppelspeicher* (→ Bild 27).

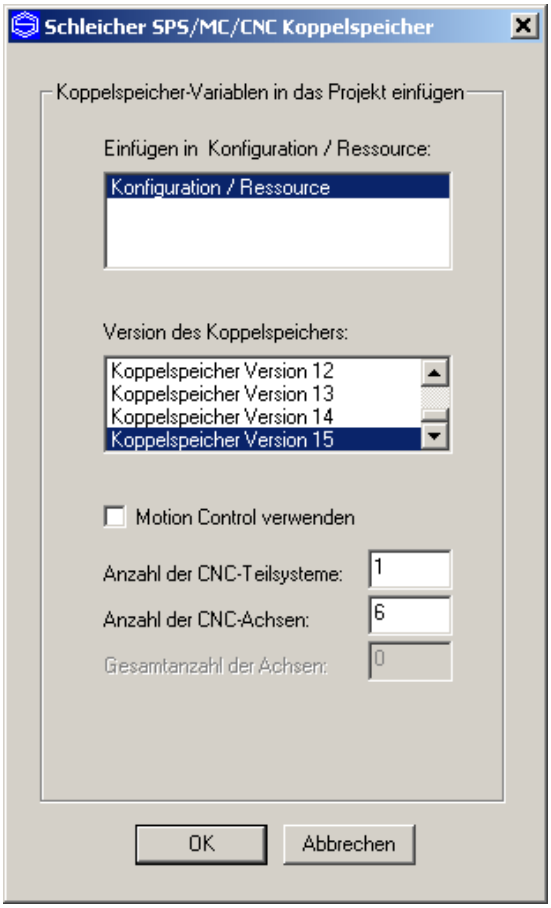
Bild 27: Aufruf des Dialogfensters "SPS/MC/CNC Koppelspeicher"



Es öffnet sich ein Dialogfenster mit folgenden Auswahl- bzw. Eingabemöglichkeiten (→ Bild 28).

- *Einfügen in Konfiguration / Ressource*:
  Auswahl der Ressource des SPS-Projektes, in die die Koppelspeicher-Variablen eingefügt werden sollen.
- *Version des Koppelspeichers*:
  Auswahl der Koppelspeicher-Version.
- *Anzahl der CNC-Teilsysteme / CNC-Achsen*:
  Eingabe der Anzahl der Teilsysteme und Achsen für CNC-Steuerungen (für reine SPS-Steuerungen sind diese beiden Eingabefelder deaktiviert).
- *OK-Button*:
  Beim Verlassen des Dialogfensters mit OK werden die Koppelspeicher-Datentypen und -Variablen in das SPS-Projekt eingefügt.
- *Abbrechen-Button*:
  Beim Verlassen des Dialogfensters mit Abbrechen wird das SPS-Projekt nicht verändert.

Bild 28: Dialogfenster "SPS/CNC/MC Koppelspeicher"



Wichtig!

Die gewählte Koppelspeicher-Version muss zur Version des Steuerungsbetriebssystems passen (siehe "Hinweise zur Auswahl der Koppelspeicher-Version", Seite 47.)

Nach dem Einfügen der Koppelspeicher-Datentypen und -Variablen muss das SPS-Projekt neu übersetzt und zur Steuerung übertragen werden.

6.7.1 Zugriff auf den Koppelspeicher

Das SPS-Programm hat Zugriff auf den gesamten Koppelspeicher über die globale Variable `plcMem` (bei SPS-Steuerungen) bzw. `cncMem` (bei CNC-Steuerungen). Mit der Punktschreibweise kann auf die einzelnen Komponenten des Koppelspeichers zugegriffen werden. Zum Beispiel kann das SPS-Programm die Versionsnummer der Betriebssoftware der Steuerung folgendermaßen auslesen: `cncMem.plcSect.lOSVersion`.

Visualisierungssysteme haben Zugriff auf den Koppelspeicher über die OPC-Schnittstelle. Die Versionsnummer der Betriebssoftware kann beispielsweise aus der OPC-Variablen `cmpS_lOSVersion` ausgelesen werden.

6.7.2 Hinweise zur Auswahl der Koppelspeicher-Version

Die Koppelspeicher-Struktur wird von Schleicher von Zeit zu Zeit aktualisiert bzw. erweitert. Zur Unterscheidung der einzelnen Varianten dient eine Versionsnummer. Die Versionsnummer wird bei größeren Änderungen an der Koppelspeicher-Struktur erhöht, bei der eine Änderung von Adressen von Variablen erforderlich ist. Koppelspeicher-Varianten mit unterschiedlicher Versionsnummer sind daher inkompatibel. Nur bei übereinstimmender Versionsnummer ist eine Kompatibilität gewährleistet.

Grundsätzlich sollte immer die aktuellste Koppelspeicher-Version verwendet werden. Wenn es sich jedoch um eine Steuerung mit einer älteren Betriebssoftware handelt, muss eine hierzu passende ältere Koppelspeicher-Version verwendet werden. Das Dokument "XCx 8xx Revision History" gibt Auskunft darüber, welche Koppelspeicher-Version mit welcher Betriebssoftware-Version verwendet werden kann. Das Dokument ist auf der Schleicher-Homepage http://www.schleicher-electronic.com unter "Betriebsanleitungen XCx" verfügbar oder wird auf Anfrage bereitgestellt.

Die Version der Betriebssoftware der Steuerung kann im Info-Dialogfenster (3) zu der entsprechenden Ressource im SPS-Projekt angezeigt werden (über (1) *Online / Projektkontrolle*, dann (2) *Ressource / Info*) (→ Bild 29).

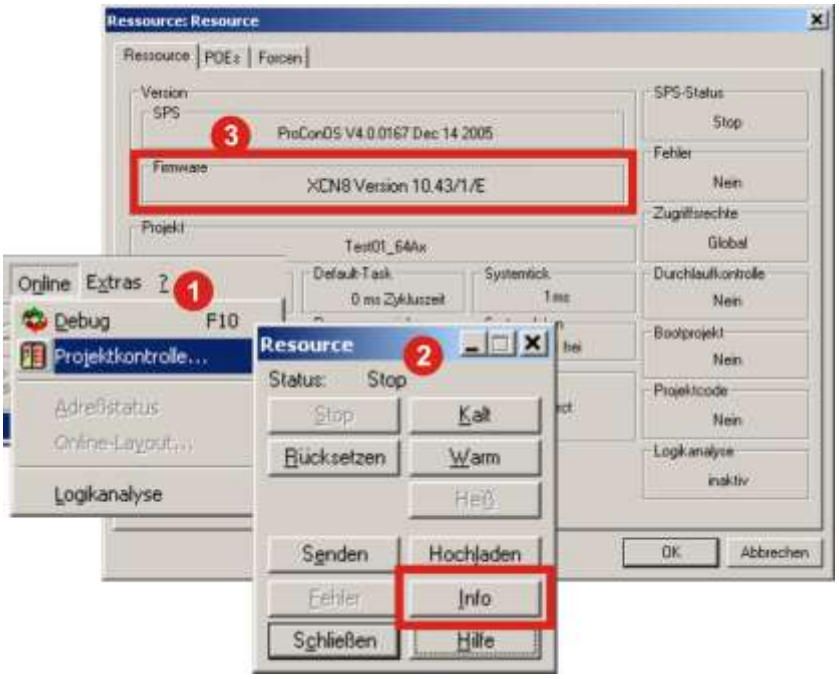
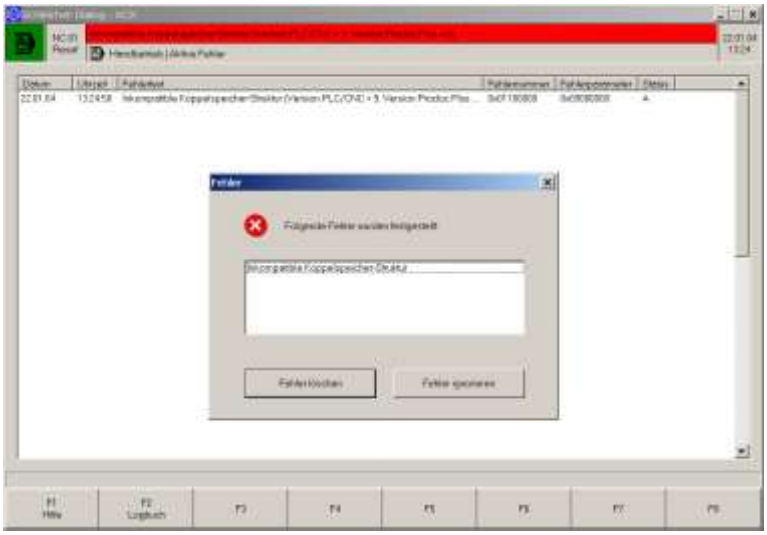


Bild 29: Auslesen der Betriebsoftware-Version

Wenn die Betriebssoftware der Steuerung und die verwendete Koppelspeicher-Version nicht übereinstimmen, wird beim Start der SPS eine Fehlermeldung in den Fehlerspeicher eingetragen. Diese Fehlermeldung wird im Schleicher Dialog folgendermaßen angezeigt (→ Bild 30).

Bild 30: Darstellung der Koppelspeicher-Fehlermeldung im Schleicher-Dialog



Falls eine solche Fehlermeldung auftritt, muss das SPS-Projekt korrigiert werden, indem die Datentypen und Variablen für die passende Koppelspeicher-Version eingefügt werden. Anschließend muss das SPS-Projekt neu übersetzt und zur Steuerung übertragen werden.

6.8 Zugriff auf die I/O-Ebene

Mit den Steuerungen XCx 400 und XCx 800 kann als I/O-Ebene eine XRIO- und / oder eine XUIO-Konfiguration erstellt werden

Mit der Installation der Add-Ons für MULTIPROG wird ein Konfigurator-Werkzeug zur Verfügung gestellt, das folgende Aufgaben übernimmt:

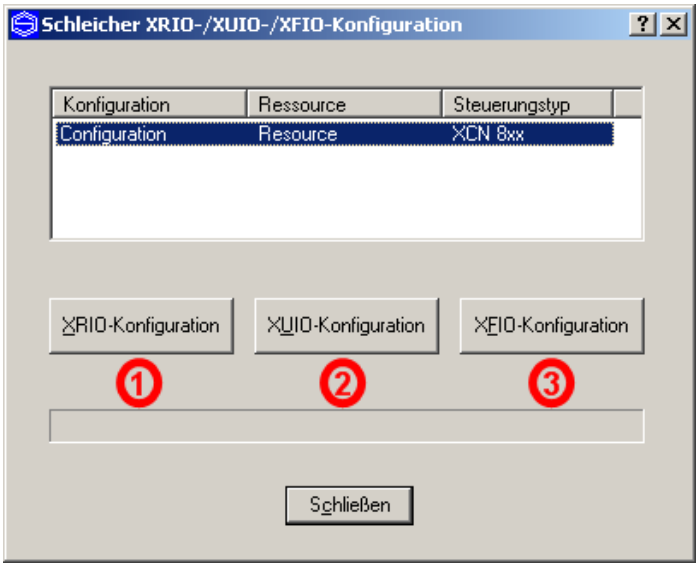
- Anzeigen und Editieren der Hardware-Konfiguration (Steckplatzliste der Eingangs- und Ausgangsmodule)
- Einfügen der erforderlichen Variablen in das SPS-Projekt, mit deren Hilfe der SPS-Programmierer auf die Eingangs- und Ausgangsmodule zugreifen kann
- Einfügen der erforderlichen Einträge in die I/O-Konfiguration des SPS-Projektes

Der Konfigurator kann über den Menüeintrag *Extras / XIO Konfiguration* gestartet werden (→ Bild 31).

Bild 31: Menüeintrag "Extras / XIO Konfiguration"

Zunächst erscheint das Dialogfenster *Schleicher XRIO-/XUIO-/XFIO-Konfiguration*, in dem die im SPS-Projekt vorhandenen Ressourcen und die zugehörigen Steuerungstypen zur Auswahl angezeigt werden. Mit dem Button *XRIO-Konfiguration* (1), Button *XUIO-Konfiguration* (2) und Button *XFIO-Konfiguration* (3) wird die Konfigurierung für die ausgewählte Ressource fortgesetzt (→ Bild 32).

Bild 32: Dialogfenster "XRIO-Konfiguration / XUIO-Konfiguration"



6.8.1 Die Auswahl im Dialogfenster XRIO-, XUIO- bzw. XFIO-Konfiguration

(1) Nach dem Aktivieren von XRIO in der XFIO-Konfiguration wird mit der Anwahl der XRIO-Konfiguration die Modulliste erstellt.

(2) Mit der Anwahl der XUIO-Konfiguration wird die Modulliste der XBUS-Module erstellt.

(3) Mit der Anwahl der XFIO-Konfiguration werden die Funktionen der I/O-Klemmen X10 und X11 festgelegt.

In jedem angewählten Dialogfenster sind die Buttons *Online-Funktionen*, *OK* und *Abbrechen* vorhanden und nur für dieses Dialogfenster wirksam..

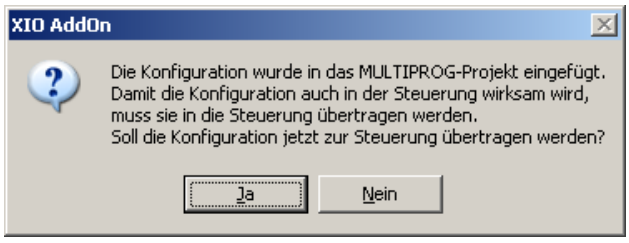
Online-Funktionen

Diese Funktion wird nur ausgeführt, wenn eine Verbindung zur Steuerung hergestellt werden kann. Diese Funktionen werden nur für die jeweilige Konfiguration vom aktiven Editor wirksam.

- **Konfiguration holen**
  Die aktuelle Konfiguration der Steuerung wird von MULTIPROG importiert und angezeigt.
  Rot markierte Module in der Modul-Konfiguration zeigen eine Differenz von Ist- zur Soll- Konfiguration in der Steuerung. Eine Differenz zur Modul-Konfiguration in MULTIPROG wird damit nicht angezeigt. Mit *OK* wird die in MULTIPROG vorhandene Konfiguration durch eingelesene Konfiguration ersetzt.
- **Konfiguration senden**
  Die in MULTIPROG eingetragene XFIO-Konfiguration oder Modul -Konfiguration wird zur Steuerung übertragen. Diese Funktion wird nur ausgeführt, wenn eine Verbindung zur Steuerung hergestellt werden kann. Die erstellte Konfiguration wird auf dem CompactFlash der Steuerung gespeichert. Die vorherige Konfiguration wird dadurch überschrieben. Nach Neustart der Steuerung ist die gesendete Konfiguration aktiv.
- **Konfiguration löschen**
  Konfiguration auf der Steuerung löschen. Diese Funktion wird nur ausgeführt, wenn eine Verbindung zur Steuerung hergestellt werden kann. Hiermit wird die auf dem CompactFlash-Speicher der Steuerung gespeicherte Modul -Konfiguration gelöscht.

OK

Abspeichern der erstellten Konfiguration, Einfügen der Variablen und I/O-Konfigurations-Einträge in das SPS-Projekt entsprechend der erstellten Modul-Konfiguration und Schließen des Fensters. Es erfolgt folgende Abfrage


XIO AddOn
Die Konfiguration wurde in das MULTIPROG-Projekt eingefügt.
Damit die Konfiguration auch in der Steuerung wirksam wird,
muss sie in die Steuerung übertragen werden.
Soll die Konfiguration jetzt zur Steuerung übertragen werden?
Ja
Nein

Diese wird mit Ja bestätigt, damit wird die Einstellung an die Steuerung übertragen und mit folgender Meldung bestätigt.

XIO AddOn
Die Konfiguration wurde in die Steuerung übertragen.
Die neue Konfiguration wird erst nach dem Neustart
der Steuerung wirksam.
OK

Ein Neustart der Steuerung kann auch nach der Übertragung weiterer Einstellungen erfolgen. Die XFIO-Konfiguration wird auch im Schleicher Dialog angezeigt.

Abbrechen

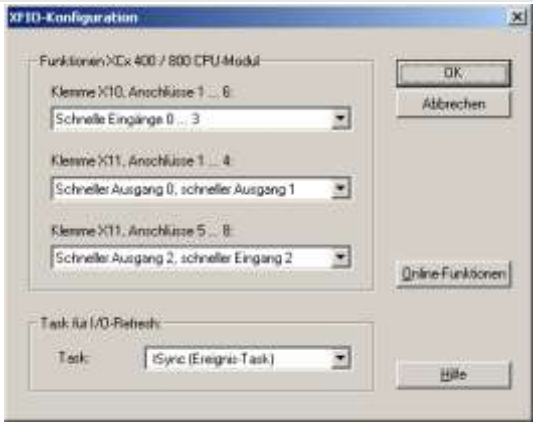
Schließen des Fensters ohne Veränderungen am SPS-Projekt und ohne Speichern der erstellten Modul -Konfiguration.

6.8.1.1 XFIO-Konfiguration (3)

Nach Aufruf des Fensters wird zunächst die im SPS-Projekt gespeicherte Konfiguration angezeigt.

I/O-Konfigurationen der Klemmen X10 und X11.

Tabelle 22: XFIO-Auswahl

XFIO-Konfiguration (3)	
Klemme X10 Anschlüsse 1...6	
Klemme X11 Anschlüsse 1...4	
Klemme X11 Anschlüsse 5...8	
Task für I/O-Refresh	

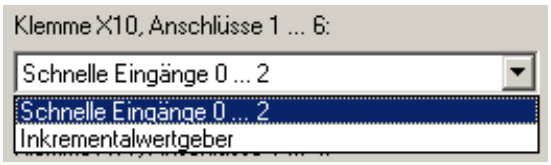
Mit bestätigt durch *OK* der gewählten Eingangs- bzw. Ausgangs-Funktion werden in *Global_Variables* in der Gruppe *XFIO_Variables* Variable dieser Funktionen angelegt und in Klartext beschrieben.

Ein Neustart der Steuerung kann auch nach der Übertragung weiterer Einstellungen erfolgen. Die aktuelle XFIO-Konfiguration wird auch im Schleicher Dialog angezeigt.

Klemme X10, Funktionsauswahl

Klemmenbelegung siehe X10 – High-Speed Eingänge 5V DC Seite 19

Bild 33: Klemme X10, Funktionsauswahl

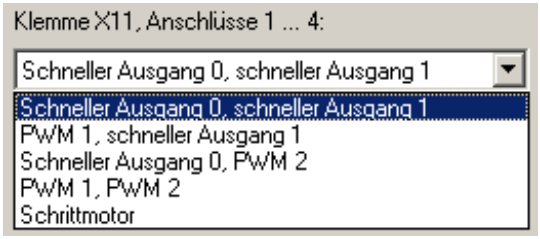


- Schnelle Eingänge
  Die schnellen Eingänge sind als 5V Differentialeingänge ausgelegt.
- Inkrementalwertgeber
  Inkrementalwertgeberanschluss mit 5V-Pegel, Beispiel siehe Inkrementalwertgeber Seite 66.

Klemme X11, Funktionsauswahl

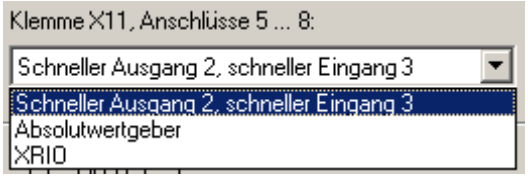
Klemmenbelegung siehe X11 – High-Speed Ausgänge 5V DC Seite 19

Bild 34: Klemme X11, Anschlüsse 1-4, Funktionsauswahl


Klemme X11, Anschlüsse 1 ... 4:
Schneller Ausgang 0, schneller Ausgang 1
Schneller Ausgang 0, schneller Ausgang 1
PWM 1, schneller Ausgang 1
Schneller Ausgang 0, PWM 2
PWM 1, PWM 2
Schrittmotor

- Mögliche Auswahl der Funktionen der schnellen Ausgänge 0 und 1.
  Die schnellen Ausgänge sind als 5V Differentialausgänge ausgelegt.
  - Schneller Ausgang 0 und schneller Ausgang 1 (siehe X9, X10, X11 Bitbelegung: Seite 65)
  - PWM 1 und schneller Ausgang 1
  - Schneller Ausgang 0 und PWM 2
  - PWM 1 und PWM 2 1 (siehe X9, X10, X11 Bitbelegung: Seite 65) (PWM = **P**uls**W**eiten**M**odulation). Beispiel siehe PWM1/2 mit den schnellen Ausgängen 0/1 Seite 72).
- Schrittmotor
  Mit dieser Einstellung kann 1 Schrittmotor angesteuert werden. Die Schrittmotordaten werden im Schleicher Dialog eingestellt.

Bild 35: Klemme X11, Anschlüsse 5-8, Funktionsauswahl


Klemme X11, Anschlüsse 5 ... 8:
Schneller Ausgang 2, schneller Eingang 3
Schneller Ausgang 2, schneller Eingang 3
Absolutwertgeber
XRIO

- Schneller Ausgang 2 und schneller Eingang 3
  Die schnellen Ein- und Ausgänge sind als 5V Differential Ein-Ausgänge ausgelegt (siehe X9, X10, X11 Bitbelegung: Seite 65).
- Absolutwertgeber
  Absolutwertgeberanschluss mit 5V- Signalpegel (siehe X9, X10, X11 Bitbelegung: Seite 65).

- XRIO
  Anschluss für RIO I/O-Module mittels XRIO- Buskoppler (siehe XRIO-Konfiguration (1)Seite 53).

Hinweis

Sonderfall: wird eine vorhandene XRIO-Konfiguration nicht für jede Anwendung benötigt und soll ausgeblendet werden, dann muss XRIO für Klemme 11 abgewählt werden.

Die vorhandene XRIO-Modulliste bleibt im MULTIPROG-Projekt und in der Steuerung erhalten.

Task für I/O-Refresh

Hiermit erfolgt die Taskzuordnung für die Ein- und Ausgänge an Klemme X9, für den Betriebsartenschalter und den Reset-Taster.

6.8.1.2 XRIO-Konfiguration (1)

Der Editor der XRIO-Konfiguration kann erst aufgerufen werden, wenn in der XFIO-Konfiguration für Klemme X11 *XRIO* eingetragen ist.

Hier kann eine Konfiguration erstellt oder von der Steuerung mit der Online-Funktion eingelesen werden (siehe Online-Funktionen Seite 50).

Vor dem Einlesen einer Konfiguration muss die *XRIO*-Einstellung für Klemme 11 in der Steuerung eingetragen und ein Steuerungsreset ausgeführt worden sein.

Die erste eingelesene Konfiguration nach der Freigabe für XRIO wird sofort wieder in die Steuerung zurück geschrieben. Damit ist die Konfiguration für den weiteren Betrieb in der Steuerung gespeichert.

Tabelle 23: XRIO- Auswahl

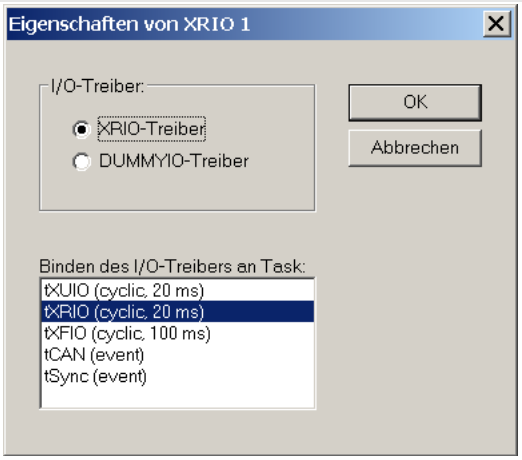
XRIO-Konfiguration (1)
Karte hinzufügen
Buskoppler hinzufügen
Modul hinzufügen
Löschen
Eigenschaften
Online-Funktionen nur für die XRIO-Konfiguration

Karte hinzufügen

Mit der Funktion Karte hinzufügen erfolgt die Zuordnung des Treibers zur auszuführenden Task. In einem neuen Template wird die vordefinierte Task *tXRIO* angeboten. Es ist nur 1 Karte zulässig.

Der *XRIO-Treiber* ist angewählt.

Tabelle 24: XRIO-Eigenschaften

XRIO-Eigenschaften	
I/O-Treiber	
Binden des I/O-Treibers an Task	

Buskoppler hinzufügen

Es können bis zu 16 Buskoppler hinzugefügt werden. Pro Buskoppler können bis zu 8 Module angefügt werden. Die Anzahl der Module kann sich durch den insgesamt verfügbaren Adressbereich von maximal 127 Byte verringern. Zur Berechnung der Byteanzahl siehe Tabelle der RIO-Modultypen und Modulklassen Seite 64.

Modul hinzufügen

Es erscheint eine Auswahlliste mit allen RIO-Modulen.

Löschen

Löschen eines oder mehrerer Module. Wenn in der Baumansicht ein Modul ausgewählt wurde, wird dieses Modul aus der Konfiguration entfernt. Wenn in der Baumansicht ein Buskoppler ausgewählt wurde, wird dieser mit allen zugehörigen Modulen entfernt.Wenn in der Baumansicht eine XRIO-Karte ausgewählt wurde, wird diese mit allen zugehörigen Buskopplern und Modulen entfernt.Wenn in der Baumansicht der Knoten XRIO-Konfiguration ausgewählt wurde, werden sämtliche XRIO-Karten, Buskoppler und Module entfernt.

Eigenschaften

Anzeigen der Eigenschaften der *XRIO-Karte*, es wird eine Auswahl der möglichen *Tasks* zur Anwendung angezeigt.

6.8.1.3 Erweiterte Funktionalität der XRIO-Konfiguration: optionale Buskoppler und optionale Module

Hinweis

Die erweiterte Funktionalität der XRIO-Konfiguration mit optionalen Buskopplern / Modulen steht SPS-Projekten unter MULTIPROG Version 5.x für die Steuerungstypen XCx 400 und XCx 800 zur Verfügung

Die Möglichkeit, Module als "optional" zu kennzeichnen, wie sie bei der XUIO-Konfiguration vorhanden ist, gibt es bei XRIO nicht. Beim Hinzufügen oder Entfernen von RIO-Modulen verschieben sich die Adressen aller nachfolgenden Module. Eine flexible Handhabung unterschiedlicher Konfigurationen ist damit schwierig. Um eine solche auch bei XRIO besser zu unterstützen, wurde die XRIO-Konfiguration um optionale Buskoppler und optionale Module erweitert.

Diese erweiterte Funktionalität steht für SPS-Projekte unter MULTIPROG Version 5.x für die Steuerungstypen XCx 400 und XCx 800 zur Verfügung und umfasst folgende Punkte.

- Es gibt für den Benutzer eine Möglichkeit zum Aktivieren bzw. Deaktivieren von Buskopplern und Modulen:
  - ☑ Buskoppler / Modul ist vorhanden
  - ☐ Buskoppler / Modul ist nicht vorhanden
- Die Namen der generierten Variablen einer festgelegten Maximalkonfiguration ändern sich durch das Aktivieren bzw. Deaktivieren von Buskopplern und Modulen nicht. Auch für die deaktivierten Module werden Variablen ohne Adressangabe erzeugt. Die Adressen werden vom Konfigurator für alle aktivierten Module aufsteigend ohne Lücken vergeben.
- Für Digitalmodule werden zusätzlich zu den bisher generierten Variablen vom Typ BYTE auch Bitvariablen vom Typ BOOL erzeugt.
- Zusätzlich wird für jeden Buskoppler eine Variable (Bitmaske vom Typ DWORD) erzeugt und mit einem entsprechenden Anfangswert initialisiert, aus dem das SPS-Programm ermitteln kann, welche Buskoppler und Module aktiviert bzw. deaktiviert sind. Für den ersten Buskoppler heißt diese Variable "xrio1_bc1", für den zweiten Buskoppler "xrio1_bc2" usw. Damit kann das SPS-Programm flexibel auf fehlende (deaktivierte) Module reagieren. Den Aufbau dieser Variablen beschreibt die Tabelle unterhalb der beiden folgenden Abbildungen.

Bild 36: XRIO-Konfiguration, Standardfunktionalität

Bild 37: XRIO-Konfiguration, erweiterte Funktionalität

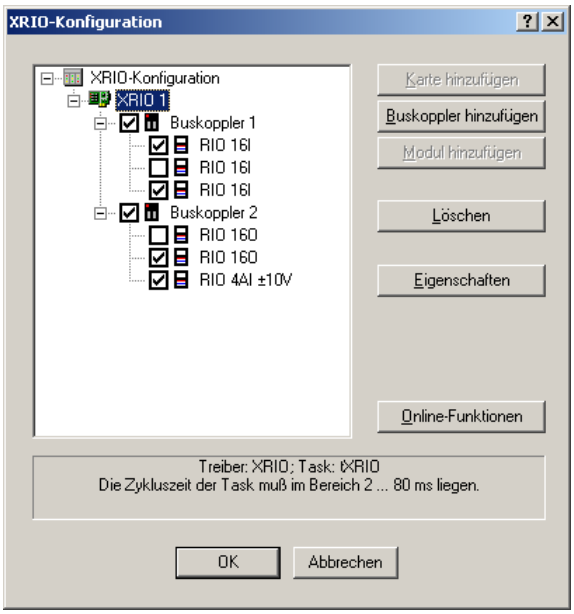


Tabelle 25 xrio1_bc<n>

Bit	Beschreibung
0	0: Buskoppler <n> deaktiviert
	1: Buskoppler <n> aktiviert
1	0: Modul 1 an Buskoppler <n> deaktiviert
	1: Modul 1 an Buskoppler <n> aktiviert
2	0: Modul 2 an Buskoppler <n> deaktiviert
	1: Modul 2 an Buskoppler <n> aktiviert
3	0: Modul 3 an Buskoppler <n> deaktiviert
	1: Modul 3 an Buskoppler <n> aktiviert
usw.	usw.

<n> = Nummer des Buskopplers, die Zählung beginnt mit 1

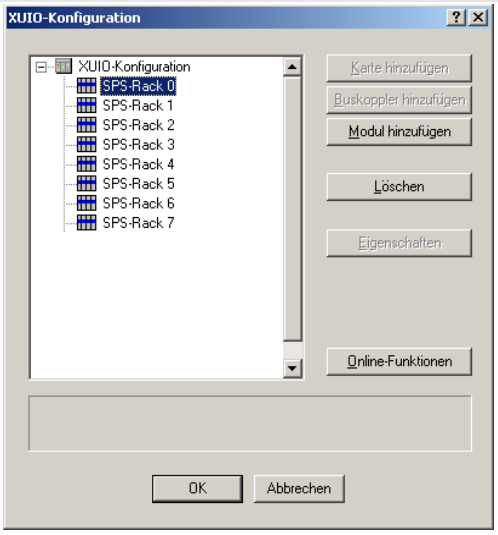
Infolge der Erweiterung des XRIO-Konfigurators muss der SPS-Programmierer folgende Punkte bei der Erstellung (Änderung) des SPS-Programms beachten:

- Am Anfang sollte einmalig eine Maximalkonfiguration erstellt werden, die sich dann möglichst nicht mehr ändert. Das nachträgliche Einfügen bzw. Löschen von Buskopplern / Modulen sollte die Ausnahme sein, da sich dadurch die Variablennamen der nachfolgenden Module ändern, so dass unter Umständen größere Anpassungen des SPS-Programms erforderlich sind.
- Die Anpassung des SPS-Projektes an eine konkrete Maschine besteht in der Regel darin, die Häkchen im Konfigurator entsprechend der physikalisch vorhandenen Buskoppler und Module zu setzen. Alle physikalisch vorhandenen Module müssen aktiviert werden (Häkchen setzen ☑) und alle physikalisch nicht vorhandenen Module müssen deaktiviert werden (Häkchen wegnehmen ☐).
  Das SPS-Programm sollte auf nicht vorhandene Module "intelligent" reagieren, indem diese nicht angesprochen werden. Hierfür ist der SPS-Programmierer verantwortlich.
- Das SPS-Programm sollte auf die RIO-Module nur anhand der vom Konfigurator generierten Variablen zugreifen. Direkte Zugriffe auf die I/O-Adressen (z.B. %QX9002.5 := true;) sind nicht mehr erlaubt. Vorhandene SPS-Programme sind entsprechend anzupassen.
- Nach dem Aktivieren bzw. Deaktivieren von Buskopplern / Modulen im Konfigurator muss
  (a) die geänderte Steckplatzliste zur Steuerung übertragen werden,
  (b) die Steuerung aus- und wieder eingeschaltet werden sowie
  (c) das SPS-Programm neu kompiliert und zur Steuerung übertragen werden.

6.8.1.4 XUIO-Konfiguration (2)

Im Dialogfenster wird die im SPS-Projekt gespeicherte Hardware-Konfiguration angezeigt. Bei einem neuen SPS-Projekt ist die Hardware-Konfiguration beim erstmaligen Öffnen dieses Fensters zunächst ohne Module. Hier kann eine Modul-Konfiguration editiert oder von einem vorhandenen Steuerungsaufbau eingelesen werden. Zum Einlesen einer Modul-Konfiguration siehe Einlesen der Hardware-Konfiguration Seite 59.

Tabelle 26: XUIO- Auswahl

XUIO-Konfiguration (2)	
Modul hinzufügen	
Löschen	
Eigenschaften	
Online-Funktionen nur für die XUIO-Konfiguration	

Modul hinzufügen

Ein neues Modul kann hinzugefügt werden, wenn ein SPS-Rack oder ein anderes Modul angewählt wird. Mit der Anwahl eines Moduls kann in einem weiteren Dialogfenster die Eigenschaften des Moduls festgelegt werden (siehe Eigenschaften Seite 58). Nach Bestätigung mit *OK* wird das Modul entsprechend der gewählten Steckplatznummer in die Modulreihe eingefügt.

Löschen

Löschen eines oder mehrerer Module. Mit Anwahl eines Moduls, wird dieses Modul aus der Hardware-Konfiguration entfernt. Mit Anwahl eines SPS-Racks, werden alle zu diesem Rack gehörigen Module entfernt. Mit Anwahl der *XUIO-Konfiguration*, werden sämtliche Module aus der Hardware-Konfiguration entfernt.

Eigenschaften

Anzeigen der Eigenschaften der *XUIO-Konfiguration* bzw. eines Moduls.

Mit Anwahl von Knoten *XUIO-Konfiguration*, kann die Darstellung der I/O-Variablen in den Global_Variables eingestellt und eine Task zugewiesen werden.

Mit Anwahl eines Moduls, werden die Eigenschaften dieses Moduls angezeigt. Hier können Steckplatz, Option und Moduladressen geändert werden (siehe Einstellen von Optionen in der Hardware-Konfiguration Seite 61).

Einlesen der Hardware-Konfiguration

Wichtig!

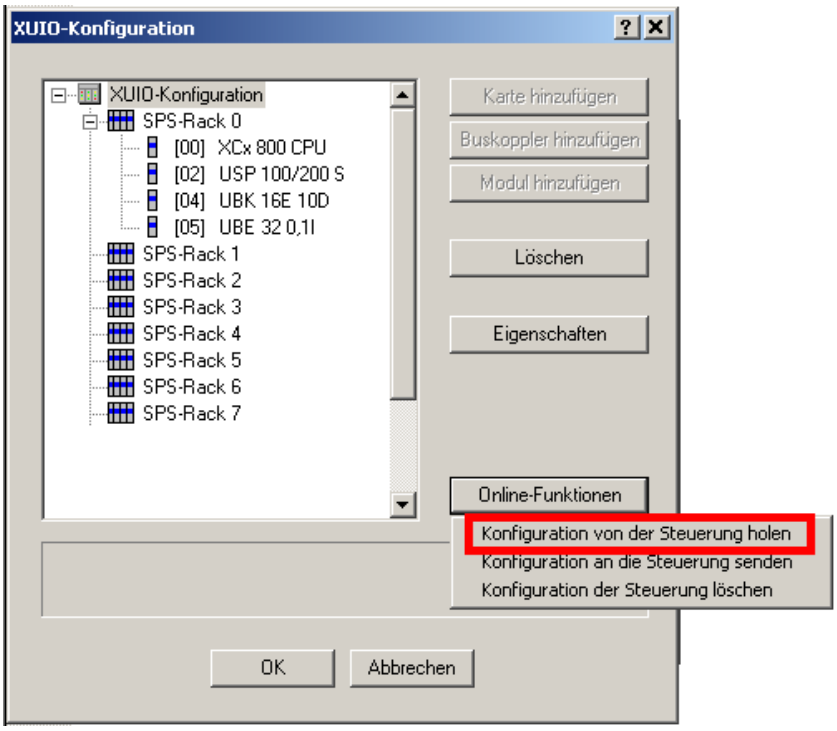
Bei Fehler in der Kommunikation über die Ethernet-Verbindung zur XCx, muss die korrekte Parametrierung der Schnittstelle in MULTIPROG überprüft werden.

In diesem Fall kann die Hardware-Konfiguration aus der mit Modulen bestückten Steuerung ausgelesen (Button *Konfiguration holen*) werden.

Zeigt die eingelesene Modul-Konfiguration Rot markierte Module mit der Meldung *Fehler in der Hardware-Konfiguration*, dann ist die von der Steuerung erkannte Modul-Konfiguration abweichend von der gespeicherten Modul-Konfiguration. Ist die eingelesene Modul-Konfiguration gültig, kann sie als gültige Modul-Konfiguration sofort in die Steuerung zurück geschrieben werden. Damit ist die Modul-Konfiguration für den weiteren Betrieb in der Steuerung gespeichert.

Mit bestätigen von *OK* wird die Modul-Konfiguration in MULTIPROG übernommen (→ Bild 38).

Bild 38: Dialogfenster "XUIO-Konfiguration"



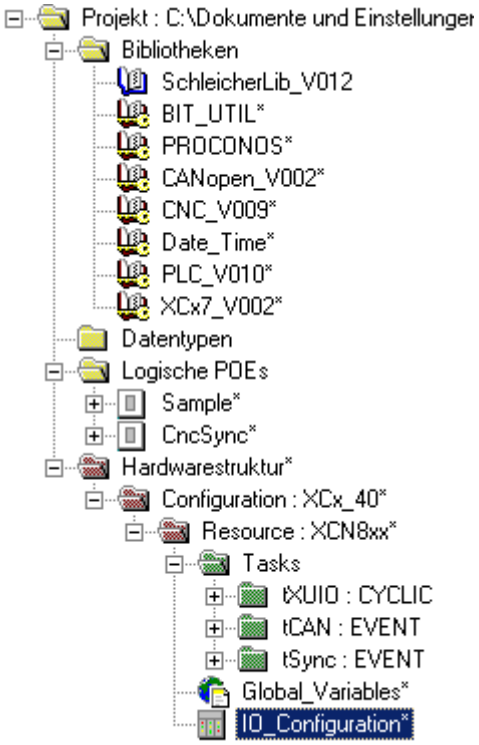
Die gespeicherte Hardware-Konfiguration wird bei jedem Steuerungsanlauf mit der tatsächlich vorhandenen Hardware-Konfiguration verglichen. Treten dabei Differenzen auf, wird der Steuerungsanlauf mit einer Fehlermeldung unterbrochen. Die LED *BUS* auf der Steuereinheit blinkt rot, im Active-Error-Buffer wird eine Fehlermeldung gespeichert:

0x01100001	Inkompatible Hardware-Konfiguration

Active-Error-Buffer und Error-Log-Book sind im Schleicher Dialog auf jeder Bedienebene über die Tastenkombination <Ctrl+?> einsehbar.

Die für die erstellte Hardware-Konfiguration erforderlichen Einträge in die I/O-Konfiguration werden in das SPS-Projekt eingefügt (die I/O-Konfiguration kann durch Doppelklick auf den Knoten *IO_Configuration* im Projektbaum geöffnet werden, → Bild 39).

Bild 39: Knoten "IO_Configuration"

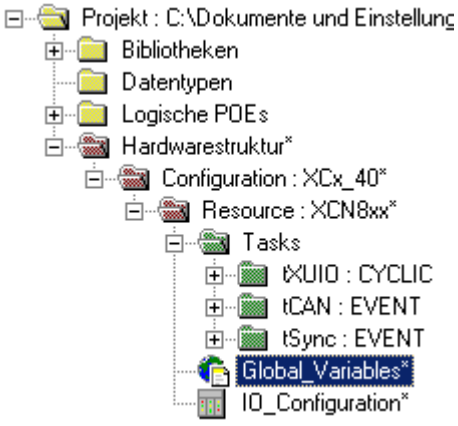


Für das oben gezeigte Beispiel werden die Einträge *XUIO_1_In* (unter INPUT) und *XUIO_1_Out* (unter OUTPUT) in die I/O-Konfiguration eingefügt (→ Bild 40).

Bild 40: Dialogfenster "I/O-Konfiguration"

Die Variablen-Tabelle für die erstellte I/O-Konfiguration wird in das SPS-Projekt eingefügt (→ Bild 42). Die Variablen-Tabelle kann durch Doppelklick auf den Knoten *Global_Variables* im Projektbaum geöffnet werden (→ Bild 41).

Bild 41: Knoten "Global_Variables"



	Name	Typ	Adresse	Beschreibung	Verwendung	Anfangsw...
	Global_Variables					
	XFIO_Variables					
	XUIO_Variables					
	xuio01_IW0	WORD	%IW 0	XUIO slot 01, UBK 16E 10D, bit inputs	VAR_GLOBAL	
	xuio01_QW0	WORD	%QW 0	XUIO slot 01, UBK 16E 10D, bit outputs	VAR_GLOBAL	
	XRIO_Variables					
	xrio1IB9000	BYTE	%IB 9000	XRIO card 1, module 1, bc 1, RIO 8I/O, digita...	VAR_GLOBAL	
	xrio1IB9001_deb	BYTE	%IB 9001	XRIO card 1, module 1, bc 1, RIO 8I/O, digita...	VAR_GLOBAL	
	xrio1QB9000	BYTE	%QB 9000	XRIO card 1, module 1, bc 1, RIO 8I/O, digita...	VAR_GLOBAL	
	xrio1QB9001_dummy	BYTE	%QB 9001	XRIO card 1, module 1, bc 1, RIO 8I/O, digita...	VAR_GLOBAL	

Bild 42: Einfügen der globalen Variablen in das SPS-Projekt

Für das obige Beispiel wurden in diesem Fall für die digitalen Ein- und Ausgänge Variablen vom Typ WORD erstellt. Dabei entspricht jedes der 16 Bits einer XUIO-Variablen einem digitalen Eingang bzw. Ausgang. Die XUIO-Variablen können jedoch auf Wunsch für jeden digitalen Ein- bzw. Ausgang auch eine eigene Variable vom Typ BOOL generiert werden. Hierzu muss die Option *Generierung von BOOL-Variablen für digitale Ein-/Ausgänge* in den Eigenschaften der XUIO-Konfiguration aktiviert werden.

Einstellen von Optionen in der Hardware-Konfiguration

Hinweis

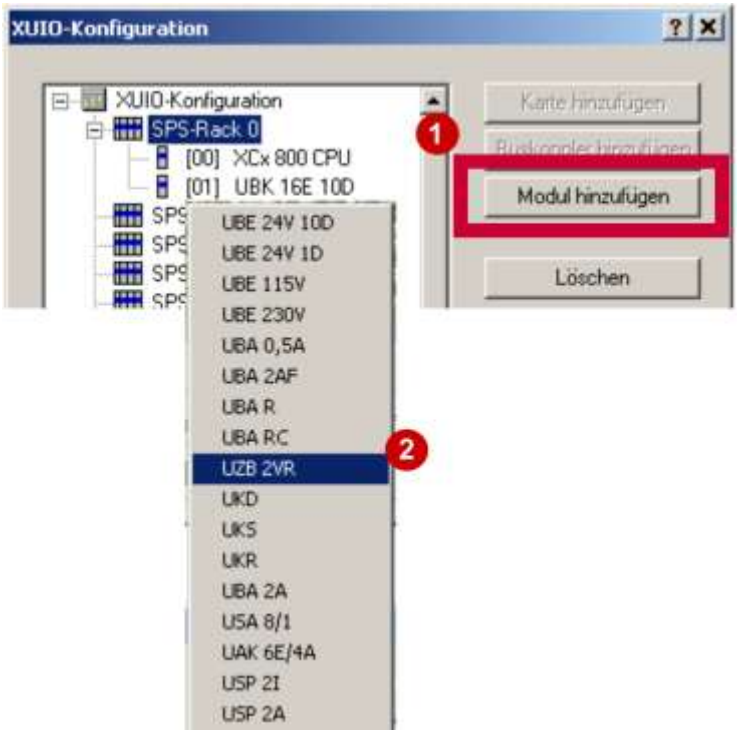
Die folgenden Ausführungen sind nur für fortgeschrittene Benutzer wichtig.

Optionen der Hardware-Konfiguration (im Folgenden Optionen genannt), erlauben eine effektive SPS-Programmerstellung. Ein SPS-Programm kann sich durch Abfrage der Optionen ohne Programmänderung an unterschiedliche Hardware-Konfigurationen anpassen.

Beispiel

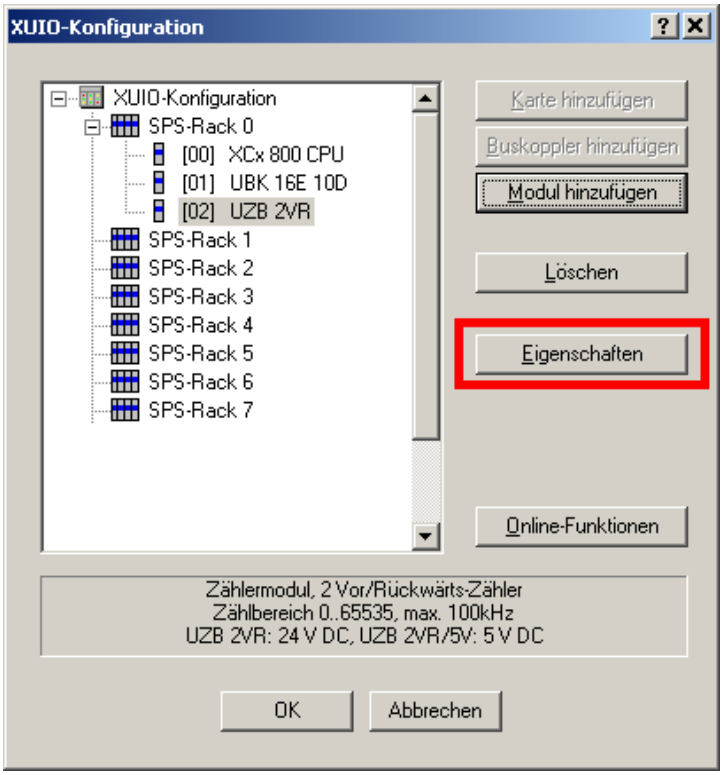
Der I/O-Konfigurator wird erneut über den Menüeintrag *Extras / XIO Konfiguration* gestartet. Dann wird die oben eingelesene Hardware-Konfiguration über den Button *Modul hinzufügen* (1) mit zwei Zähler-Modulen UZB 2VR ergänzt (2) (→ Bild 43).

Bild 43: Dialogfenster "XUIO-Konfiguration", Modul hinzufügen



Anschließend wird für jedes der beiden neuen Module über den Button *Eigenschaften* das Dialogfenster *Modul-Eigenschaften* aufgerufen (→ Bild 44).

Bild 44: Dialogfenster "XUIO-Konfiguration", Eigenschaften



Für beide Module wird die Einstellung *Option* im Dialog *Modul-Eigenschaften* auf 1 eingestellt (→ Bild 45). Es sind insgesamt 32 Optionen 0...31 einstellbar. Alle anderen Module behalten ihre Einstellung, voreingestellt ist Option 0.

Bild 45: Dialogfenster "Modul-Eigenschaften", Einstellen der Optionen

Die beiden UZB 2VR gehören damit zur Option 1, alle anderen Module zur Option 0.

Bei Steuerungsanlauf wird die gespeicherte Hardware-Konfiguration mit der tatsächlich vorhandenen Hardware verglichen. Von einer Option müssen dabei entweder alle Module vorhanden sein (die Option ist dann aktiv), oder es darf kein Modul dieser Option vorhanden sein (die Option ist dann nicht aktiv). Die Adressbereiche nicht aktiver Optionen bleiben reserviert.

Sind die Module einer Option nur teilweise vorhanden, wird ein Konfigurationsfehler erkannt. (LED BUS auf der Steuereinheit blinkt rot, im Active-Error-Buffer wird eine Fehlermeldung gespeichert.)

6.8.2 Tabelle der RIO-Modultypen und Modulklassen

Module- ID	HID	Name	Module Type	Input Bytes	Output Bytes	I/O- Bytes effektive
0x0001	0x02	RIO 8I/O	DIGITAL	1+1	2	3
0x0002	0x06	RIO 16I	DIGITAL	2+2	0	3
0x0003	0x04	RIO 16O	DIGITAL	0	2	3
0x0004	0x08	RIO 8I 8I/O	DIGITAL	2+2	2	3
0x0005	0x0A	RIO 4AI/4AO ±10V	ANALOG	8	8	9
0x0006	0x0C	RIO 4AI ±10V	ANALOG	8	0	9
0x0007	0x0E	RIO 4AI/4AO 20mA	ANALOG	8	8	9
0x0008	0x10	RIO 4AI 20mA	ANALOG	8	0	9
0x0009	0x12	RIO RNO	DIGITAL	2+2	0	3
0x000A	0x42	RIO C24-10	COUNTER	10+2	10	11
0x000B	0x42	RIO C24-10	COUNTER	10+2	10	11
0x000C	0x62	RIO P24-10	COUNTER	10+2	10	11
0x000D	0x62	RIO P24-10	COUNTER	10+2	10	11
0x000E	0x16	RIO T10-10	ANALOG	8	0	9
0x000F	0x1E	RIO 4AI 0..10V	ANALOG	8	0	9
0x0010	0x18	RIO 4AI/4AO 4..20	ANALOG	8	8	9
0x0011	0x1A	RIO 4AI 4..20mA	ANALOG	8	0	9
0x0012	0x1C	RIO 4AI/4AO 0..10	ANALOG	8	8	9
0x0013	0x24	RIO 4OR	DIGITAL	0	2	3
0x0014	0x20	RIO T20-10	ANALOG	8	0	9
0x0015	0x00	RIO P05-10	COUNTER	10+2	10	11
0x0016	0x00	RIO P05-10	COUNTER	10+2	10	11
0x0017	0x44	RIO 8O 2Amp.	DIGITAL	0	2	3
0x0018	0x26	RIO 4I 115VAC	DIGITAL	2	0	3
0x0019	0x46	RIO 4I 230VAC	DIGITAL	2	0	3
0x001A	0x00	RIO 8I TTL	DIGITAL	2	0	3
0x001B	0x00	RIO 8O neg	DIGITAL	0	2	3
0x001C	0x50	RIO A 10-10	ANALOG	8+4	8	9
0x001D	0x70	RIO ECX2	DIGITAL	2	2	3

Tabelle 27: Modultypen

Input Bytes sind die Summe von entprellten und nicht entprellten Eingangs-Bytes.

I/O-Bytes effektive sind die Summe aller zuschiebenden resp. zu übertragenden Bytes.

Tabelle 28: Modulklassen

Modul- Typ	Code
ANALOG	0
DIGITAL	1
COUNTER	2
NOMOD	3

6.9 SPS-Applikationsbeispiele für XCx400 / XCx800 I/O Schnittstellen

SPS-Applikationsbeispiele für die I/O-Schnittstellen X9 bis X11 der Steuereinheiten XCx400 und XCx800.

6.9.1 X9, X10, X11 Bitbelegung:

- X9)
  8I davon 4I/O für Standard I/O, Interruptfunktion (siehe SPS-Beispiel für Interrupt mit den Eingängen der XCx 400 / XCx 800 Seite 73).
- X10)
  - Schnelle Eingänge 0 - 2
    Die schnellen Eingänge sind als 5V Differentialeingänge ausgelegt (siehe schnelle Ausgänge 0-2 / schnelle Eingänge 0-3: Seite 65).
  - Inkrementalwertgeber
    Inkrementalwertgeberanschluss mit 5V-Pegel. Beispiel siehe Inkrementalwertgeber Seite 67.
- X11)
  - Schnelle Ausgänge 0 - 2 und schneller Eingang 3
    Die schnellen Ein- und Ausgänge sind als 5V Differential Ein-Ausgänge ausgelegt (siehe schnelle Ausgänge 0-2 / schnelle Eingänge 0-3: Seite 65)
  - Absolutwertgeber
    Absolutwertgeberanschluss mit 5V-Signalpegel. Beispiel siehe Absolutwertgeber Seite 69.
  - Schnelle Ausgänge 0 und 1.
    Diese schnellen Ausgänge können als PWM betrieben werden (PWM = **P**uls**W**eiten**M**odulation). Beispiel siehe PWM1/2 mit den schnellen Ausgängen 0/1 Seite 72.

6.9.2 schnelle Ausgänge 0-2 / schnelle Eingänge 0-3:

Variablen in für die Einstellung schnelle Ausgänge 0-2 / schnelle Eingänge 0-3.

Variable	Wert	Datentyp	Adresse	Beschreibung
xfio_IW0	16#0800	WORD	%IW 10000	XFIO, inputs (read)
xfio_QW0	16#0000	WORD	%QW 10000	XFIO, outputs (read + write)
xfio_IB2	16#03	BYTE	%IB 10002	XFIO, key switch
xfio_IB3	16#00	BYTE	%IB 10003	XFIO, reset key
xfio_MW0	16#8080	WORD	%MW 3.6000000	XFIO, I/O status (read + write)
xfio_MW4	16#0800	WORD	%MW 3.6000004	XFIO, inputs (read)
xfio_MW6	16#0000	WORD	%MW 3.6000006	XFIO, outputs (read + write)

Tabelle 29: XFIO-Parameter

Die Ein- und Ausgänge von X9, X10, X11 werden durch die Variablen *xfio_IW0* (siehe Tabelle 30: Eingangsbits xfio_IW0 Seite 66) geschrieben und *xfio_QW0* (siehe Tabelle 31: Ausgangsbits xfio_QW0Seite 66) gelesen.

Tabelle 30: Eingangsbits xfio_IW0

xfio_IW0	**Eingänge an X9**	**Anschluss**
Bit 0	Input slow 0	Klemme X9 3 [I/O0]
Bit 1	Input slow 1	Klemme X9 4 [I/O1]
Bit 2	Input slow 2	Klemme X9 5 [I/O2]
Bit 3	Input slow 3	Klemme X9 6 [I/O3]
Bit 4	Input slow 4	Klemme X9 7 [I4]
Bit 5	Input slow 5	Klemme X9 8 [I5]
Bit 6	Input slow 6	Klemme X9 9 [I6]
Bit 7	Input slow 7	Klemme X9 10 [I7]
xfio_IW0	**Eingänge an X10, X11**	**Anschluss**
Hinweis: offene Eingänge werden als High-Signal gelesen		
Bit 8	Input fast 0	Klemmen X10 1/2 [IN0 u. /IN0]
Bit 9	Input fast 1	Klemmen X10 3/4 [IN1 u. /IN1]
Bit 10	Input fast 2	Klemmen X10 5/6 [IN2 u. /IN2]
Bit 11	Input fast 3	Klemmen X11 7/8 [IN3 u. /IN3]
Bit 12 ... 15		ohne Funktion

Tabelle 31: Ausgangsbits xfio_QW0

xfio_QW0	**Ausgänge an X9**	**Anschluss**
Bit 0	Output slow 0	Klemme X9 3 [I/O0]
Bit 1	Output slow 1	Klemme X9 4 [I/O1]
Bit 2	Output slow 2	Klemme X9 5 [I/O2]
Bit 3	Output slow 3	Klemme X9 6 [I/O3]
Bit 4...7		ohne Funktion
xfio_QW0	**Ausgänge an X11**	**Anschluss**
Bit 8	Output fast 0	Klemmen X11 1/2 [OUT0 u. /OUT0]
Bit 9	Output fast 1	Klemmen X11 3/4 [OUT1 u. /OUT1]
Bit 10	Output fast 2	Klemmen X11 5/6 [OUT2 u. /OUT2]
Bit 11 ... 15		ohne Funktion

6.9.3 Inkrementalwertgeber

Für die Inkrementalwertgeber-Funktion stehen 4 Variablen für die Auswertung zur Verfügung.

Variable	Datentyp	Beschreibung	Anwendung
xfio_MW164_INC_control	WORD	Control	read/write
xfio_MW168_INC_counter_lw	WORD	Zähler Low	read/write
xfio_MW170_INC_counter_hw	WORD	Zähler High	read/write
xfio_MW178_INC_status	WORD	Status	read

Tabelle 32: Inkrementalwertgeber -Parameter

- *xfio_MW164_INC_control,* mit dieser Variablen wird die Nullimpulsauswertung eingestellt.

Tabelle 33: xfio_MW164_INC_control

Bit	Beschreibung
0	0: Nullsignalauswertung ist deaktiviert
	1: Nullsignalauswertung ist aktiviert, mit der 1. steigenden Flanke vom Nullsignal wird der Zähler auf 0 gesetzt.
1	0. Nullsignal nicht speichern, gespeichertes Nullsignal löschen
	1: Nullsignal speichern
2-15	Ohne Funktion

Wichtig!

Das Lesen bzw. Schreiben der Counter-Variablen muss immer paarweise erfolgen. Es muss erst die Variable *xfio_MW168_INC_counter_lw,* dann die Variable *xfio_MW170_INC_counter_hw* gelesen bzw. geschrieben werden.

Das Lesen bzw. Schreiben der Variablen *xfio_MW168_INC_counter_lw* als DWORD ist nicht zulässig.

- *xfio_MW168_INC_counter_lw*, Zähler Low Word, das Auslesen bzw. Beschreiben muss vor dem High Word erfolgen.
- *xfio_MW170_INC_counter_hw*, Zähler High Word, das Auslesen bzw. Beschreiben muss nach dem Low Word erfolgen.
- *xfio_MW176_INC_type*, in Bit 0-3 wird mit 0101 angezeigt, daß die Inkrementalwertgeber-Funktion enabled ist.
- *xfio_MW178_INC_status*, Zustandsanzeige des Zählers.

Tabelle 34: xfio_MW178_INC_status

Bit	Beschreibung
0	Zustand Spursignal A
1	Zustand Spursignal B
2	Zustand Spursignal N
3	0: Nullsignal nicht aufgetreten (steigende Flanke)
	1: Nullsignal aufgetreten
4-15	Ohne Funktion

SPS-Beispiel für Inkrementalwertgeber

Name	Typ	Verwendung	Beschreibung	Adresse
xfio_MW164_INC_control	WORD	VAR_EXTERNAL	XFIO, INC ENCODER, control register	
xfio_MW168_INC_counter_lw	WORD	VAR_EXTERNAL	XFIO, INC ENCODER, counter, low word	
xfio_MW170_INC_counter_hw	WORD	VAR_EXTERNAL	XFIO, INC ENCODER, counter, high word	
xfio_MW178_INC_status	WORD	VAR_EXTERNAL	XFIO, INC ENCODER, status register	
NullImp	BOOL	VAR	Nullimpuls vorhanden	
RefNeu	BOOL	VAR	Nullimpulsspeicher löschen zum Referieren	
RefFahrt	BOOL	VAR	Referenzfahrt mit Nullimpuls speichern	
CountRd	BOOL	VAR	Geberwert lesen	
CountWr	BOOL	VAR	Geberwert schreiben	
CountValRd	DWORD	VAR	Geberwert gelesen	
CountValWr	DWORD	VAR	Geberwert schreiben	
LowWordRd	WORD	VAR	Kopie von xfio_MW168_INC_counter_lw	%MW 0
HighWordRd	WORD	VAR	Kopie von xfio_MW170_INC_counter_hw	%MW 2
LowHighWordRd	DWORD	VAR		%MD 0
LowWordWr	WORD	VAR	Kopie an xfio_MW168_INC_counter_lw	%MW 4
HighWordWr	WORD	VAR	Kopie an xfio_MW170_INC_counter_hw	%MW 6
LowHighWordWr	DWORD	VAR		%MD 4

Tabelle 35 SPS-Variablenliste Inkrementalwertgeber

```
(* Beispiel Inkrementalgeber auswerten *)
(* 1. Nullimpuls lesen *)
NullImp:= WORD_TO_BOOL(xfio_MW178_INC_status and WORD#8);
(* 1. Nullimpuls aktivieren und Speichern *)
IF RefFahrt = TRUE THEN
    xfio_MW164_INC_control := WORD#3;
    RefFahrt:= FALSE;
END_IF;
(*  Nullimpulsspeicher löschen *)
IF RefNeu = TRUE THEN
    IF NullImp = TRUE THEN
       xfio_MW178_INC_status:= WORD#16#0000;
    END_IF;
END_IF;
(* Geberwert als DWORD lesen *)
IF CountRd = TRUE THEN
    LowWordRd := xfio_MW168_INC_counter_lw;
    HighWordRd := xfio_MW170_INC_counter_hw;
    CountValRd := LowHighWordRd;
END_IF;
(* Geberwert als DWORD Schreiben *)
IF CountWr = TRUE THEN
    LowHighWordWr := CountValWr ;
    xfio_MW168_INC_counter_lw := LowWordWr ;
    xfio_MW170_INC_counter_hw := HighWordWr ;
END_IF;
```

6.9.4 Absolutwertgeber

Für die Absolutwertgeber-Funktion stehen 4 Variablen für die Auswertung zur Verfügung.

Variable	Datentyp	Beschreibung	Anwendung
xfio_MW228_ABS_control	WORD	Control	read/write
xfio_MW232_ABS_value_lw	WORD	Zähler Low	read
xfio_MW234_ABS_value_hw	WORD	Zähler High	read
xfio_MW242_ABS_status	WORD	Status	read

Tabelle 36: Absolutwertgeber-Parameter

- *xfio_MW228_ABS_control,* mit dieser Variablen werden die Geberdaten eingestellt.

Tabelle 37: xfio_MW228_ABS _control

Bit	Beschreibung								
0	0:								
	1: SSI Einlesevorgang starten								
1	0. Gray-Code								
	1: Binär-Code								
	Taktfrequenz								
2	0	100 kHz	1	200 kHz	0	500 kHz	1	1 MHz	
3	0		0		1		1		
4...7	Ohne Funktion								
8	Datenbreite einstellen, z.B. 24 Bit-Geber			Bit 0, LSB		0			
9				Bit 1		0			
10				Bit 2		0			
11				Bit 3		1			
12				Bit 4		1			
13				Bit 5, MSB		0			
14, 15	Ohne Funktion								

Wichtig!

Das Lesen der Counter-Variablen muss immer paarweise erfolgen. Es muss erst die Variable *xfio_MW232_ABS_value_lw,* dann die Variable *xfio_MW234_ABS_value_hw* gelesen werden.

Das Lesen der Variablen *xfio_MW232_ABS_value_lw* als DWORD ist nicht zulässig.

- *xfio_MW232_ABS_value_lw*, Zähler Low Word, das Auslesen bzw. Beschreiben muss vor dem High Word erfolgen.
- *xfio_MW234_ABS_value_hw*, Zähler High Word, das Auslesen bzw. Beschreiben muss nach dem Low Word erfolgen.
- *xfio_MW242_ABS_status*, Zustandsanzeige vom SSI-Protokoll.

Tabelle 38: xfio_MW242_ABS_status

Bit	Beschreibung
0	0: SSI Einlesevorgang abgeschlossen
	1: SSI Einlesevorgang läuft
1	0: Gray-Code
	1: Binär-Code
2	Abbild der eingestellten Taktfrequenz
3	
4	0: SSI Daten nicht invertiert
	1: SSI Daten invertiert
5	0: SSI Takt nicht invertiert
	1: SSI Takt invertiert
6	Ohne Funktion
7	0: kein SSI Fehler
	1: SSI Fehler, mögliche Ursache Fehler der eingestellten Datenbreite Fehler der eingestellten Taktfrequenz Fehler im Protokoll
8	Abbild der eingestellten Datenbreite
9	
10	
11	
12	
13	
14, 15	Ohne Funktion

SPS-Beispiel für Absolutwertgeber

Name	Typ	Verwendung	Beschreibung	Adresse
xfio_MW228_ABS_control	WORD	VAR_EXTERNAL	XFIO, ABS ENCODER, control register	
xfio_MW232_ABS_value_lw	WORD	VAR_EXTERNAL	XFIO, ABS ENCODER, value, low word	
xfio_MW234_ABS_value_hw	WORD	VAR_EXTERNAL	XFIO, ABS ENCODER, value, high word	
xfio_MW242_ABS_status	WORD	VAR_EXTERNAL	XFIO, ABS ENCODER, status register	
LowWordRd	WORD	VAR	Kopie von xfio_MW232_ABS_counter_lw	%MW 0
HighWordRd	WORD	VAR	Kopie von xfio_MW234_ABS_counter_hw	%MW 2
LowHighWordRd	DWORD	VAR		%MD 0
CountValRd	DWORD	VAR	Geberwert gelesen	
EinlesReady	BOOL	VAR	Einlesevorgang abgeschlossen	
SSI_Error	BOOL	VAR		
Err	BOOL	VAR		

Tabelle 39 SPS-Variablenliste Absolutwertgeber

Wichtig!

Die im Beispiel eingetragene Reihenfolge :

- **Abfrage des Auslesevorgangs,**
- **Abfrage auf SSI-Fehler,**
- **Lesen der Geberwerte,**
- **Starten des Einlesevorgangs**

soll eingehalten werden.

(* Absoluwertgeber *)

```
(* Die im Beispiel eingetragene Reihenfolge :
   Abfrage des Auslesevorgangs,
   Abfrage auf SSI-Fehler,
   Lesen der Geberwerte und
   Starten des Einlesevorgangs
soll eingehalten werden.
Bei Programmstart kann ein SSI-Fehler auftreten, die
erste Statusauswertung erfolgt vor dem Einlesevorgang. *)

(* Bitabfrage mit dem BIT_TEST aus der Gruppe BIT_UTIL *)
(* Bit 0 abfragen auf Einlesevorgang fertig *)

EinlesReady:= BIT_TEST(xfio_MW242_ABS_status,sint#0);

(* Bit 7 abfragen auf SSI Fehler *)

SSI_Error:= BIT_TEST(xfio_MW242_ABS_status,sint#7);

(* Fehlerbehandlung *)

IF SSI_Error = true THEN
 Err:=true;
END_IF;

(* Geberwert lesen *)

IF EinlesReady = FALSE THEN
 LowWordRd := xfio_MW232_ABS_value_lw;
 HighWordRd := xfio_MW234_ABS_value_hw;
 CountValRd := LowHighWordRd;

(* SSI Einlesevorgang mit den Absolutgeberdaten, 24 Bit,
100 kHz, und starten mit Bit 0 = 1 *)

 xfio_MW228_ABS_control:= WORD#16#1801;
END_IF;
```

6.9.5 PWM1/2 mit den schnellen Ausgängen 0/1

Die PWM-Funktion wird durch 3 Variable konfiguriert.
(PWM = **P**uls**W**eiten**M**odulation)

Variablen PWM1	Datentyp	Beschreibung	Anwendung
xfio_MW180_PWM_freq	UINT	Grundeinheit	read/write
xfio_MW182_PWM_high_dur	UINT	High-Dauer	read/write
xfio_MW184_PWM_low_dur	UINT	Low-Dauer	read/write
Variablen PWM2	**Datentyp**	**Beschreibung**	**Anwendung**
xfio_MW196_PWM_freq	UINT	Grundeinheit	read/write
xfio_MW198_PWM_high_dur	UINT	High-Dauer	read/write
xfio_MW200_PWM_low_dur	UINT	Low-Dauer	read/write

Tabelle 40: PWM-Variablen

Variablenfunktion für PWM1, identisch zu PWM2:

- *xfio_MW180_PWM_freq*, diese Varable ist die Grundeinheit zur Erstellung der High und Low –zeiten. Sie kann zwischen 1*20ns und 65535 *20ns eingestellt werden. Mit dem Wert = 0 ist die PWM-Funktion abgeschaltet.
- *xfio_MW182_PWM_high_dur*, die Highzeit wird berechnet durch (n * xfio_MW180_PWM_freq). Mit dem Wert = 0 ist die PWM-Funktion abgeschaltet. Die kürzeste Highzeit darf 500ns nicht unterschreiten.
- *xfio_MW184_PWM_low_dur*, die Lowzeit wird berechnet durch (n * xfio_MW180_PWM_freq). Mit dem Wert = 0 und Grundeinheit <> 0 wird ein Highpegel ausgegeben. Die kürzeste Lowzeit darf 500ns nicht unterschreiten.
- *xfio_MW192_PWM_type*, in Bit 0-3 wird mit 0011 angezeigt das PWM enabled ist.

Berechnungsbeispiel für eine Periodenzeit von 1ms:

- Die Grundeinheit soll 1µs (50 * 20ns) sein, damit wird der Wert 50 in die Variable *xfio_MW180_PWM_freq* eingetragen.
- Die Highzeit soll 400µs (400 * Grundeinheit 1µs) sein, damit wird der Wert 400 in *xfio_MW182_PWM_high_dur* eingetragen.
- Die Lowzeit soll 600µs (600 * Grundeinheit 1µs) sein, damit wird der Wert 600 in *xfio_MW184_PWM_low_dur* eingetragen.

Mit diesem Konfigurationsbeispiel können Änderungen der High bzw Low –zeit mit einer Schrittweite von 1µs erfolgen. Die Änderung der Schrittweite kann nur durch Wertänderung in der Variablen *xfio_MW180_PWM_freq* erfolgen. Soll die High und Low –zeit erhalten bleiben, dann müssen die Werte der Variablen *xfio_MW180_PWM_freq* und *xfio_MW184_PWM_low_dur* neu berechnet werden.

SPS-Beispiel für PWM

Das Programmbeispiel ist für PWM1 und kann in der Anwendung auf PWM2 übertragen werden.

Name	Typ	Verwendung	Beschreibung	Anfangswert
xfio_MW180_PWM_freq	UINT	VAR_EXTERNAL	XFIO, PWM 1, frequency	
xfio_MW182_PWM_high_dur	UINT	VAR_EXTERNAL	XFIO, PWM 1, high duration	
xfio_MW184_PWM_low_dur	UINT	VAR_EXTERNAL	XFIO, PWM 1, low duration	
bPWM1Start	BOOL	VAR	PWM aktiv mit TRUE	
PWM1Freq	UINT	VAR	Grundeinheit = n*20 ns	50
PWM1High	UINT	VAR	Highzeit = n*Grundeinheit	400
PWM1Low	UINT	VAR	Lowzeit = n*Grundeinheit	600

Tabelle 41 SPS-Variablenliste PWM

```
(* PWM1 Schalten 0.4µs ein und 0.6µs aus*)

IF bPWM1Start = TRUE THEN
   xfio_MW180_PWM_freq:= PWM1Freq;
   xfio_MW182_PWM_high_dur:= PWM1High;
   xfio_MW184_PWM_low_dur:= PWM1Low;
END_IF;
```

6.9.6 SPS-Beispiel für Interrupt mit den Eingängen der XCx 400 / XCx 800

Die XCx 800 kann bis zu 12 Interrupts verarbeiten. Diese können durch die digitalen Eingänge der Schnittstellen X9 (Seite 19), X10 (Seite 19) und X11 (Seite 19) ausgelöst werden.

Interruptfunktion im CNC-Betrieb mit $53 / $54

- Für sercos III sind die Eingänge I0 und I1 von X9 der XCx800 zu benutzen.
- Für die Achsmodule auf dem X-Bus / U-Bus sind die ersten vier Eingänge eines Eingangs-Modules UBE 32 0,1I bzw. eines XBE 32 0,1 I zu benutzen (siehe Betriebsanleitung 'Erweiterungsmodule für XCx', BestellNr. R4.322.2400.0).

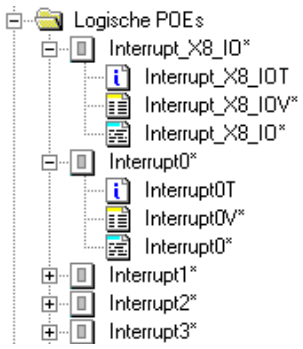
6.9.6.1 Interruptverarbeitung mit dem Funktionsbaustein XFIO_CONFIG

Die digitalen Eingänge der Schnittstellen X9, X10 und X11 der XCx 400 / XCx 800 können mittels dem Funktionsbaustein XFIO_CONFIG der SPS als Interrupteingänge ausgewertet werden. (→ Programmbeispiel POE " Interrupt_X8_IO " Seite 74).

6.9.6.2 Beispiel-POEs für die Interruptverarbeitung

Es werden drei POEs erstellt und in die Taskstruktur eingebunden. Die POEs dienen sowohl der Interrupt-Freigabe *Interrupt_X8_IO* als auch der Interrupt-Verarbeitung eines Messtasters *Interrupt 0* und eines Zählimpulses *Interrupt1* (→ Bild 46).

Bild 46:
Knoten für Interrupt-freigabe und -verarbeitung



Es werden globale Variablen angelegt für die Auswertung der aufgetretenen Interrupts und für die Kommunikation der POEs untereinander (→ Bild 47).

Bild 47:
Globale Variablen für Interruptzählung und Kommunikation

Interrupt0_Freigabe	BOOL	VAR_GLOBAL
Interrupt0	BOOL	VAR_GLOBAL
Interrupt1 _Zaehler	DINT	VAR_GLOBAL
Zielposition	DINT	VAR_GLOBAL

POE " Interrupt_X8_IO "

Interrupt mit den XCx400 / XCx800 Eingängen (X9, X10, X11)

Beispiel 1
Interrupt für XCN800 mit sercos III: steigende Flanke, Single-Mode, Eingang I0, Interrupt0, $54 Messtaster.
Auswertung der ersten steigender Flanke am Eingang (z.B. ein Messtaster). Die Interrupt0_RTrig_Freigabe muss hier manuell durch Forcen gesetzt werden. Sie wird dann im Programm Interrupt0 zurückgesetzt

Name	Typ	Verwendung	Beschreibung	Adresse
XFIOenable	BOOL	VAR	TRUE für Interruptfreigabe	FALSE
IOchannel	BYTE	VAR	I0 von X9	0
XFIOmode	BYTE	VAR	enable sercos interrupt, Interrupttask Event0	2
XFIOerror	WORD	VAR		
XFIO_CONFIG_1	XFIO_CONFIG	VAR		
WRPenable	BOOL	VAR	TRUE für warten auf Interrupt	FALSE
RRPenable	BOOL	VAR		TRUE
axNr	USINT	VAR	Achse 1	1
parNr	UDINT	VAR	$54 für Meßtaster	25
wrData	DINT	VAR		1
rdData	DINT	VAR		
WRPerror	WORD	VAR		
RRPerror	WORD	VAR		
WRITE_AXIS_PAGE_1	WRITE_AXIS_PAGE	VAR		
READ_AXIS_PAGE_1	READ_AXIS_PAGE	VAR		

Tabelle 42 SPS-Variablenliste Interrupt

```
XFIO_CONFIG_1(ENABLE:=xfioEnable,
      CHANNEL:=IOchannel,
             MODE:= XFIOmode,
             R_EDGE:= FALSE,
             F_EDGE:= TRUE);
XFIOerror:=XFIO_CONFIG_1.ERROR;

(* ============ $54 init in remote page ================= *)

WRITE_AXIS_PAGE_1(ENABLE:=WRPenable,
      AXIS_NR:= axNr,
             PARA_NR:= parNr,
             WRITEDATA:= wrData);
WRPerror:=WRITE_AXIS_PAGE_1.ERROR;

READ_AXIS_PAGE_1(ENABLE:=RRPenable,
      AXIS_NR:=axNr,
             PARA_NR:=parNr);
             rdData:=READ_AXIS_PAGE_1.READDATA;
RRPerror:=READ_AXIS_PAGE_1.ERROR;
```

Zum Deaktivieren der Interrupteingänge müssen die Variablen R_EDGE und F_EDGE mit FALSE initialisiert werden.

Beispiel 2
Interrupt, fallende Flanke, Continuierlicher-Mode, Eingang I2, Interrupt1
Auswertung jeder fallenden Flanke am Eingang (zB. Durchlaufzähler),

Name	Typ	Verwendung	Beschreibung	Adresse
XFIOenable	BOOL	VAR	TRUE für Interruptfreigabe	FALSE
IOchannel	BYTE	VAR	I2 von X9	2
XFIOmode	BYTE	VAR	enable continuous mode, Interrupttask Event1	17
XFIOerror	WORD	VAR		
XFIO_CONFIG_1	XFIO_CONFIG	VAR		

```
XFIO_CONFIG_1(ENABLE:=xfioEnable,
      CHANNEL:=IOchannel,
             MODE:= XFIOmode,
             R_EDGE:=TRUE,
             F_EDGE:=FALSE);
XFIOerror:=XFIO_CONFIG_1.ERROR;
```

Zum Deaktivieren der Interrupteingänge müssen die Variablen R_EDGE und F_EDGE mit FALSE initialisiert werden.

POE "Interrupt0" (Messtater)

Beispiel 1:
POE Interrupt0/ Task Int0, Ereignis0
Interrupt0 (X9 Eingang I0)
Auswertung der ersten steigender Flanke am Eingang als Messtaster

```
(*  ------------------------------------------------------------------ *)
Zielposition := cncMem.axSect[1].wrdN2P.lOrigMeasVal;

Interrupt0_Freigabe:= FALSE;
Interrupt0 := FALSE;

RETURN;
```

POE "Interrupt1" (Zähler)

Beispiel 2:
POE Interrupt1/ Task Int1, Ereignis1
Interrupt1 (X9 Eingang I2))
Auswertung aller fallenden Flanken am Eingang als Zaehler

```
(* ------------------------------------------------------------------ *)
Interrupt1_Zaehler  := Interrupt1_Zaehler + 1;

RETURN;
```

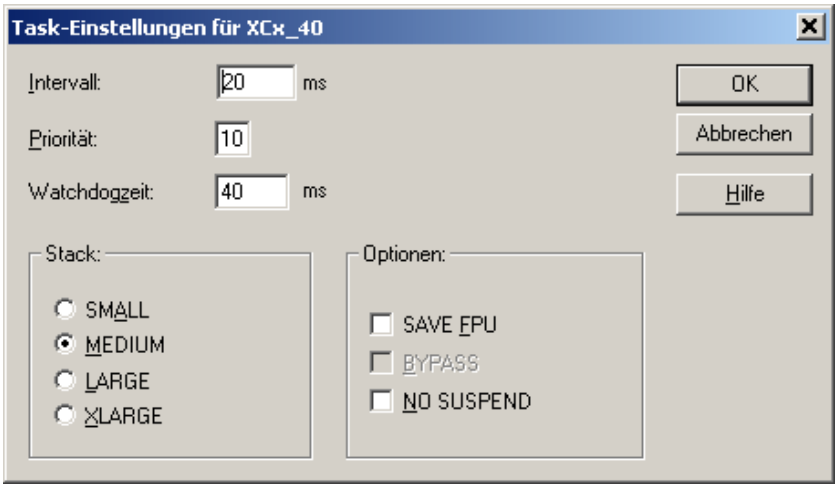
6.9.6.3 Taskstruktur für die Interruptverarbeitung

Für die Interruptfreigabe wird eine zyklische Task mit dem Namen *XUIOTsk* angelegt, der die POE *UBE32_Interrupt* zugewiesen wird (→ Bild 48).

Bild 48: Zyklische Task "XUIO" mit POE "X8_Interrupt_IO"

Als Beispiel werden folgende Task-Einstellungen gewählt (→ Bild 49):

Bild 49: Einstellungen für zyklische Task



Zuletzt werden Event-Tasks angelegt, die interruptabhängig gestartet und denen die interrupverarbeitenden POEs zugewiesen werden (→ Bild 50).

Bild 50:
Anlegen der Event-Tasks
für die Interrupts

Hier werden Event-Tasks für die Interruptauswertung angelegt:

- *Int_0:EVENT* für Interrupt über Eingangs-Bit 0 mit POE *Interrupt0*
- *Int_1:EVENT* für Interrupt über Eingangs-Bit 1 mit POE *Interrupt1*

Die Zuweisung der Interrupts (Eingangs-Bits) zu den Event-Tasks muss in der Taskeinstellung mittels der Ereignisnummer angegeben werden (→ Bild 51)

Bild 51:
Einstellungen für
Event-Tasks

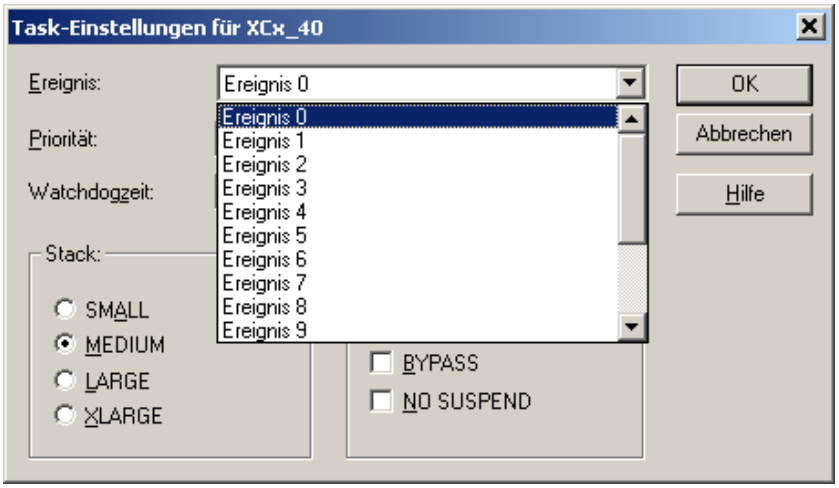
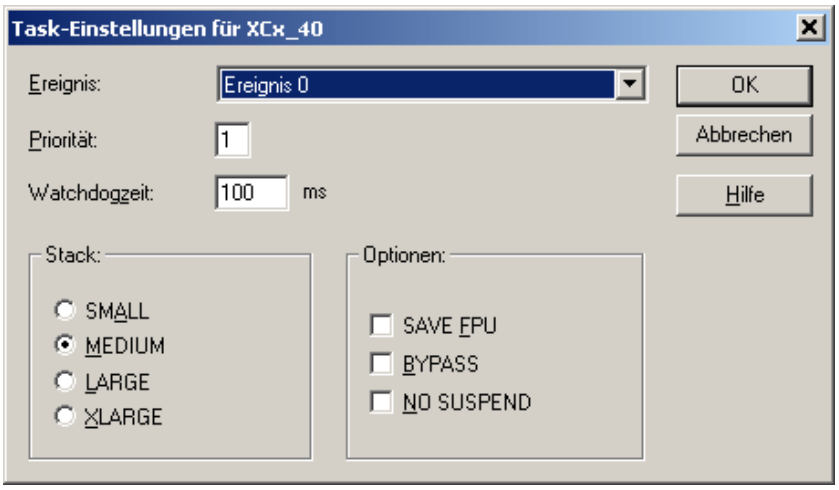


Bild 52:
Einstellungen für
Event-Tasks



6.10 CANopen für dezentrale I/O

Die XCx 800 besitzt eine integrierte CANopen-Schnittstelle für die Feldbusanbindung oder die Ansteuerung digitaler Antriebe. Dieses Kapitel beschreibt anhand einer Minimalkonfiguration die Inbetriebnahme und Konfiguration des CANopen-Netzwerkes.

6.10.1 Spezifikationen

- CANopen arbeitet mit zwei Typen von Telegrammen:
  - **SDO** (Service Data Objects) sind Telegramme, die vom Empfänger **bestätigt** werden müssen,
  - **PDO** (Process Data Objects) sind Telegramme, die vom Empfänger **nicht bestätigt** werden müssen.
- Während der Netzwerk-Konfiguration werden die PDOs für den Datenaustausch definiert und bekommen eine sogenannte COB ID. Der Empfänger einer Nachricht erkennt zu jeder Zeit, welches Telegramm für diesen Knoten bestimmt ist.
- Einige Komponenten unterstützen nur sogenanntes Default Mapping und arbeiten mit festen COB IDs, die in der CANopen-Definition festgeschrieben sind.
- Die Standardkommunikation für PDO ist **COS** (**C**hange **O**f **S**tate): Eine PDO wird nur gesendet, wenn sich die Information innerhalb der PDO ändert.

6.10.2 Anschlussprinzip und Verkabelung

Der beschriebene Aufbau ist eine Minimalkonfiguration, die für die weitere Inbetriebnahme als Beispiel dient.

Bild 53: Anschlussprinzip für CANopen-Netzwerk

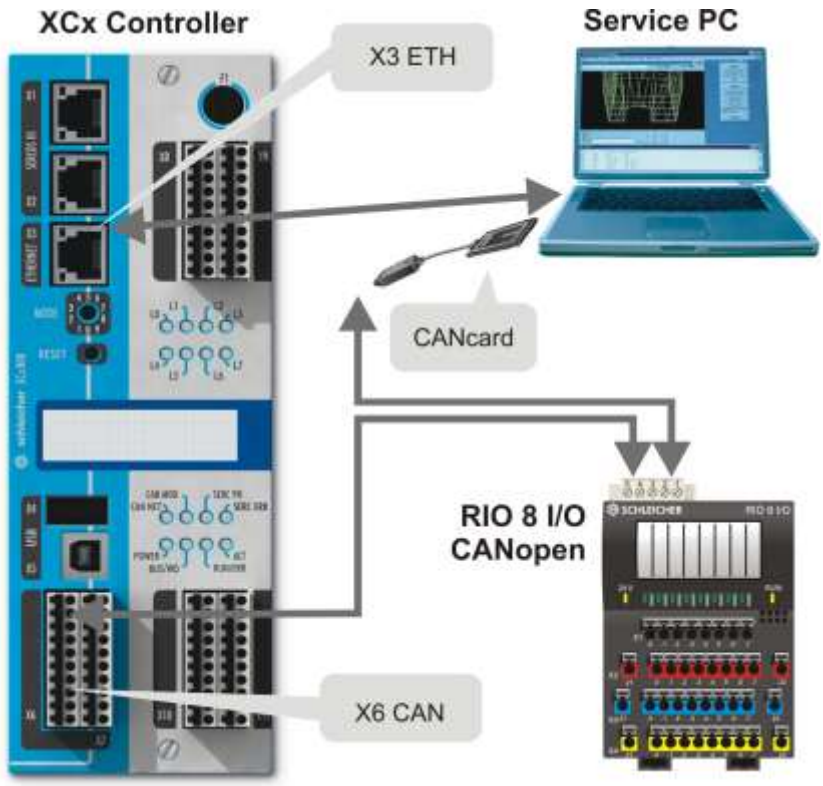
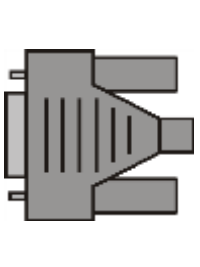


Tabelle 43: Pinbelegung der verwendeten Stecker



	1	2	3
	D-Sub, 9-polig, Buchse zur CANcard im Service-PC	Steckblock-klemme 10-pol. X6, an der XCx 800	Steckblock-klemme 5-pol. am RIO 8 I/O CANopen
CAN	**Pin**	**Pin**	**Pin**
0 V		5 / 10	1
CAN_L	2	4 / 9	2**
Drain		3 / 8	3
CAN_H	7	2 / 7	4**
DC +24 V		1 / 6	5

* Die Pingruppen 1..5 und 6..10 sind parallel geschaltet.

** Zwischen Pin 2 und 4 am RIO 8 I/O CANopen muss ein Abschluss-widerstand von 120 Ohm geschaltet werden.

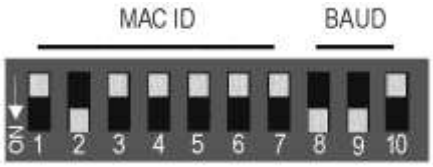
6.10.3 Einstellungen am I/O-Modul RIO 8 I/O CANopen

Am Kompaktmodul RIO 8 I/O CANopen die Knotennummer 2 und die Datenübertragungsrate 125 kBaud einstellen (→ Bild 54). Dazu den DIP-Schalter auf der Moduloberseite einstellen:

Tabelle 44: Einstellen von Knotennummer und Datenübertragungsrate

	Knotennummer MAC ID							**Datenübertragungsrate BAUD**		
Schalter	1	2	3	4	5	6	7	8	9	10
Stellung	OFF	ON	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF

Bild 54: DIP-Schalter am Kompaktmodul RIO 8 I/O CANopen



6.10.4 Deklaration des I/O-Treibers für CANopen

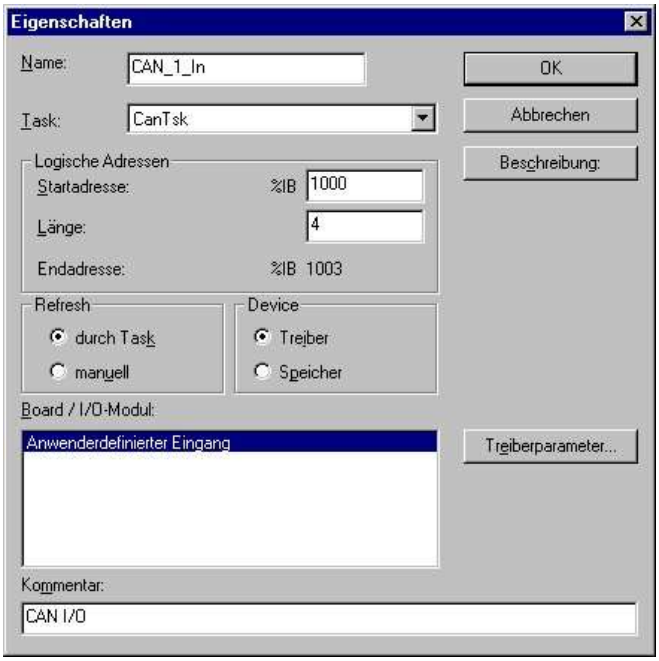
Ort und Prinzip der Deklaration des I/O-Treibers für das CANopen-Netzwerk sind dieselben wie für XUIO. Am Ende des Projektbaums befindet sich der Container *IO_Configuration*. Dort sind bereits die I/O-Konfigurationen *CAN_1_In* (unter INPUT) und *CAN_1_Out* (unter OUTPUT) erstellt (→ Bild 55).

Bild 55: CANopen-Konfiguration, Dialogfenster "I/O-Konfiguration"

Für dieses Beispiel müssen folgende Parameter eingetragen werden (→ Bild 56):

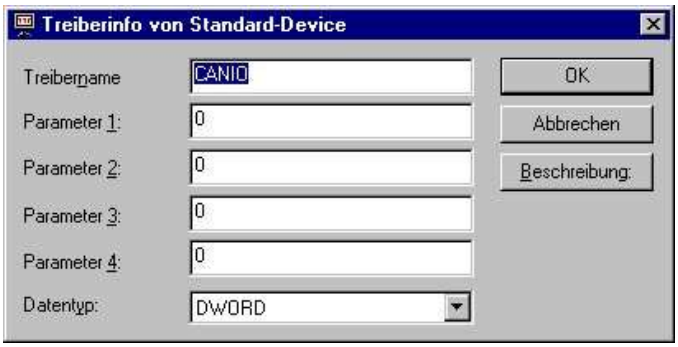
- Die **Task**, mit welcher der I/O Adressraum synchronisiert wird, muss *CanTsk* sein.
- Als **Startadresse** werden die logischen Adressen *IB1000* für *CAN_1_In* und *QB1000* für *CAN_1_Out* eingetragen.
- Im Parameter **Länge** werden so viele I/O Bytes deklariert, wie im CANopen-Netzwerk ausgetauscht werden sollen (hier 4, da minimal Doppelwortabstände angewendet werden).

Bild 56: CANopen-Konfiguration, Dialogfenster "Eigenschaften"



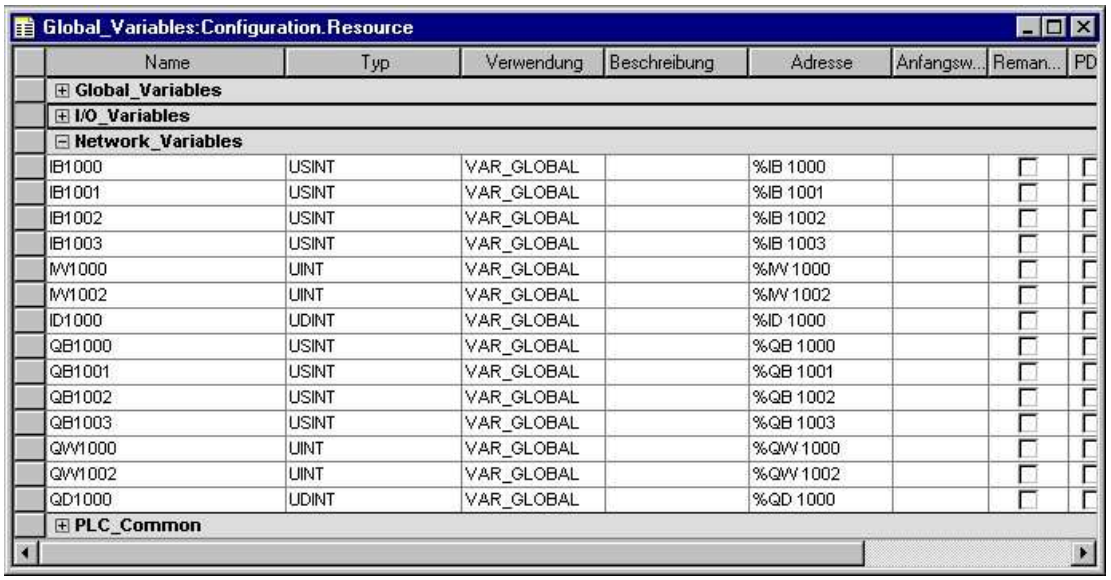
In den **Treiberparametern** muss der Treibername *CANIO* eingestellt sein, der Datentyp ist DWORD (→ Bild 57).

Bild 57: CANopen-Konfiguration, Treiberparameter einstellen



6.10.5 Deklaration von Netzwerkvariablen in MULTIPROG

Im Fenster des Projektbaums unter *Global_Variables* sind im Ordner *Network_Variables* die benötigten Variablen vordefiniert (→ Bild 58).



Global_Variables:Configuration.Resource

Name	Typ	Verwendung	Beschreibung	Adresse	Anfangsw...	Reman...	PD
⊞ Global_Variables							
⊞ I/O_Variables							
⊟ Network_Variables							
IB1000	USINT	VAR_GLOBAL		%IB 1000		☐	☐
IB1001	USINT	VAR_GLOBAL		%IB 1001		☐	☐
IB1002	USINT	VAR_GLOBAL		%IB 1002		☐	☐
IB1003	USINT	VAR_GLOBAL		%IB 1003		☐	☐
IW1000	UINT	VAR_GLOBAL		%IW 1000		☐	☐
IW1002	UINT	VAR_GLOBAL		%IW 1002		☐	☐
ID1000	UDINT	VAR_GLOBAL		%ID 1000		☐	☐
QB1000	USINT	VAR_GLOBAL		%QB 1000		☐	☐
QB1001	USINT	VAR_GLOBAL		%QB 1001		☐	☐
QB1002	USINT	VAR_GLOBAL		%QB 1002		☐	☐
QB1003	USINT	VAR_GLOBAL		%QB 1003		☐	☐
QW1000	UINT	VAR_GLOBAL		%QW 1000		☐	☐
QW1002	UINT	VAR_GLOBAL		%QW 1002		☐	☐
QD1000	UDINT	VAR_GLOBAL		%QD 1000		☐	☐
⊞ PLC_Common							

Bild 58: CANopen-Konfiguration, Arbeitsblatt "Network_Variables"

I/O-Bits werden im Arbeitsblatt *I/O_Variables* mit den Adressen *IX1000.0* und *QX1000.0* deklariert (→ Bild 59). (Im Beispiel wird *QX1000.7* benutzt, um das Ergebnis am RIO 8 I/O sichtbar zu machen.)

Global_Variables:Configuration.Resource

Name	Typ	Verwendung	Beschreibung	Adresse	An
⊞ Global_Variables					
⊟ I/O_Variables					
CAN_1_IX0	BOOL	VAR_GLOBAL	CANopen 1. Netzwerk Eingang 0	%IX 1000.0	
CAN_1_QX7	BOOL	VAR_GLOBAL	CANopen 1. Netzwerk Ausgang 7	%QX 1000.7	
XRIO_1_IX0	BOOL	VAR_GLOBAL	XRIO Modul 1 (RIO 16 I) Eingang 0	%IX 0.0	
XRIO_1_IX1	BOOL	VAR_GLOBAL	XRIO Modul 1 (RIO 16 I) Eingang 1	%IX 0.1	
XRIO_2_QX0	BOOL	VAR_GLOBAL	XRIO Modul 2 (RIO 16 O) Ausgang 0	%QX 0.0	

Bild 59: CANopen-Konfiguration, Arbeitsblatt "I/O_Variables"

6.10.6 CANopen-Konfiguration mit "Schleicher CANopen Konfiguration"

Für die Einrichtung einfacher CANopen-Netze steht die **Schleicher CANopen Konfiguration** zur Verfügung, die zusammen mit den MULTIPROG-Add-Ons installiert wird. Der Konfigurator ist ein reines Offline-Werkzeug, d.h. es besteht keine Kommunikation zur CAN-Karte der Steuerung. Das Werkzeug generiert Programmcode, der beim Start der SPS ausgeführt wird. Dabei werden eine Reihe von SDOs zur Steuerung übertragen, die das Netzwerk konfigurieren.

Bild 60:
Aufruf der "Schleicher CANopen Konfiguration" in MULTIPROG

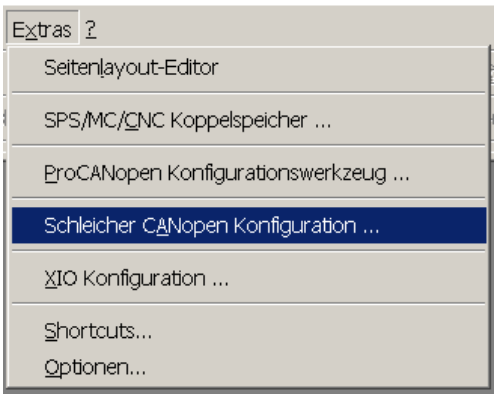
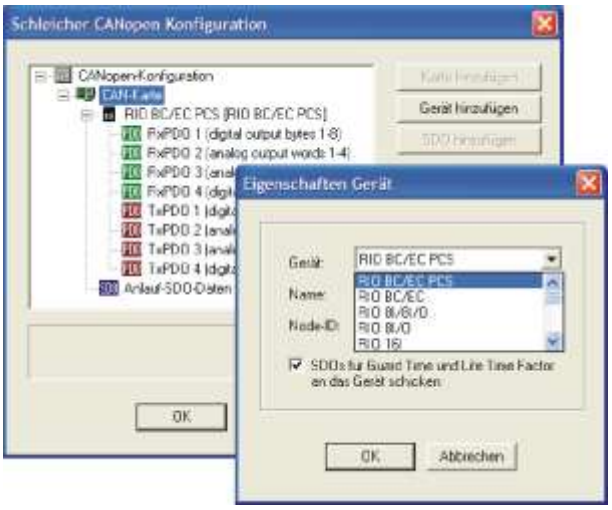


Bild 61:
Schleicher CANopen Konfiguration



Im Konfigurationsfenster können (je nach Steuerung) eine oder mehrere CAN-Karten konfiguriert werden. CAN-Geräte wie Buskoppler oder Antriebe lassen sich einfach über eine Auswahlliste hinzufügen. Dabei werden automatisch die vom Gerät unterstützten Empfangs- und Sende-PDOs eingefügt. Anschließend wird die erstellte Konfiguration gespeichert und alle erforderlichen Änderungen am SPS-Projekt (inkl. Generierung des SPS-Programmcodes) vorgenommen.

Da der Konfigurator offline arbeitet, ist keine direkte Fehlerdiagnose mit dem Werkzeug selbst möglich. Es werden jedoch SPS-Programmcode sowie Variablen erzeugt, die es erlauben, die Konfigurierung des CANopen-Netzwerkes nach dem Start der SPS zu überprüfen. Die Werte dieser Variablen können im Online-Modus von MULTIPROG überprüft werden.

6.10.7 CANopen-Konfiguration mit "ProCANopen"

ProCANopen ist ein Konfigurationswerkzeug von der Fa. Vector Informatik GmbH

Komplexe CANopen-Netzwerke können mit dem **ProCANopen Konfigurationswerkzeug** konfiguriert werden. Dazu ist die Software "ProCANopen" erforderlich, die nicht im Lieferumfang von MULTIPROG enthalten ist. Zusätzlich wird eine CAN-Feldbuskarte im Service-PC benötigt, z.B. "CANcardY" (→ Seite 134).

Die Eigenschaften und Fähigkeiten der Komponenten sind in einer EDS-Datei (**E**lectronic **D**ata **S**heet) deklariert. Die EDS-Datei muss in das Unterverzeichnis von ProCANopen mit dem Namen EDS kopiert werden.

ProCANopen bildet die (abbildbaren) Objekte der Knoten ab. Beispielsweise werden Ausgangs-Bytes der RIO Module (die die Eingangs-Bits vom RIO repräsentieren) mit Eingangs-Bytes der XCx verknüpft. Darüber hinaus werden zusätzliche Informationen für den Feldbus konfiguriert:

- Welcher Knoten ist der "NMT manager"?
- Welcher Knoten ist der "Configuration manager"?
- "Guarding" und "Guarding time"
- "Sync time" und "Sync window length"

Nach der Netzwerkkonfiguration kann diese im Netzwerk gespeichert werden. Im Netzwerk speichern bedeutet: der Knoten, welcher als "Configuration manager" ausgewählt ist (meistens XCx), bekommt via CANopen die Information, wie das Netzwerk zu konfigurieren ist. Der "Configuration manager" speichert die Information (z.B. bei der XCx auf der Flashdisk), und nach dem Einschalten konfiguriert die XCx das Netzwerk.

Nachdem das Netzwerk konfiguriert ist, kann der "NMT manager" das CAN-Netzwerk starten. Netzwerk starten bedeutet: Status "operational" und Datenaustausch von PDOs (**P**rocess **D**ata **O**bjects).

6.10.7.1 Installation von ProCANopen

Hinweis

Für Anwender, welche bereits ProCANopen Version 2.1 und MULTIPROG 1.2 für die Schleicher-Steuerungen MicroLine und ProNumeric/ProSycon installiert haben:

- **Sie benötigen nur die Update-Version von ProCANopen.**
- **Überschreiben Sie nicht die bereits installierte Version!**
- **Installieren Sie ProCANopen V3.2 in einem neuen Pfad auf der Harddisk z.B. \ProCANopen3.**
- **Wenn Sie eine CANCardX im Einsatz haben, ist es manchmal erforderlich, ein Update der Firmware und der Options auf der Karte vorzunehmen: Bitte notieren Sie die Seriennummer der Karte und setzen Sie sich mit Ihrem lokalen Händler in Verbindung.**
- **In Abhängigkeit von Ihrem PC-Betriebssystem müssen Sie verschiedene Treiber installieren. Einige neue Treiber sind nicht kompatibel mit der älteren ProCANopen-Version 2.1! Dies bedeutet, ProCANopen Version 2.1 mit dem neuen Treiber V3.x funktioniert nicht mehr online mit dem Feldbus!**
- **Mit ProCANopen V2.1 projektierte Konfigurationsdateien können mit ProCANopen V3.2 weiter benutzt werden.**

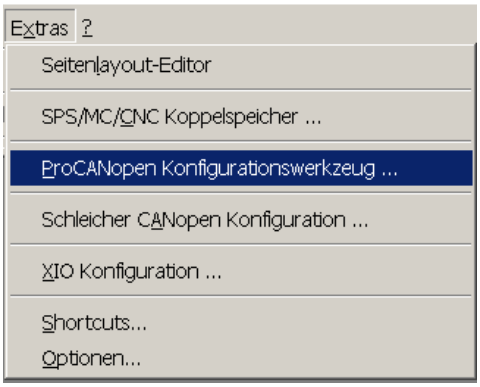
Um ProCANopen zu installieren, folgen Sie bitte der Dokumentation, die mit der Software und der CAN-Karte geliefert wurde. Sie müssen die Treiber und die ProCANopen-Software in zwei Schritten installieren:

- Kopieren sie die aktuellen EDS-Dateien für den Steuerungstyp von der Service-CD in das Verzeichnis *\...\ProCANopen\EDS*.
- Wenn die Dialogsprache Englisch gewünscht ist, ändern Sie in der Datei *\....\ProCANopen\EXE\VECTOR.INI* die Zeile "language=0049" in "language=001".

6.10.7.2 Einbindung von ProCANopen in MULTIPROG

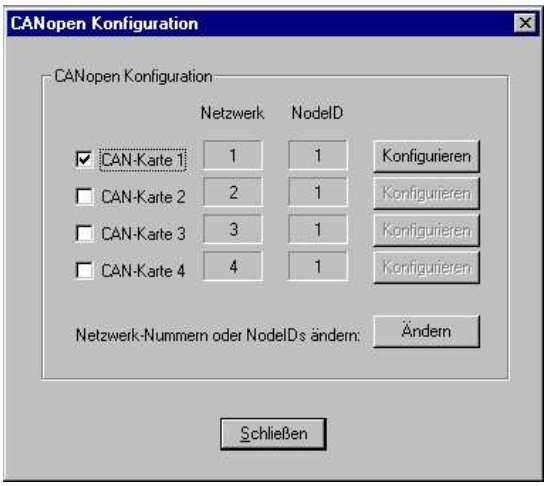
Die Installation der Add-Ons bereitet MULTIPROG so vor, dass ProCANopen direkt von MULTIPROG aus gestartet werden kann. Dazu im Menü *Extras* / *ProCANopen* Konfigurationswerkzeug wählen (→ Bild 62).

Bild 62: Aufruf von "ProCANopen Konfigurations-werkzeug" in MULTIPROG



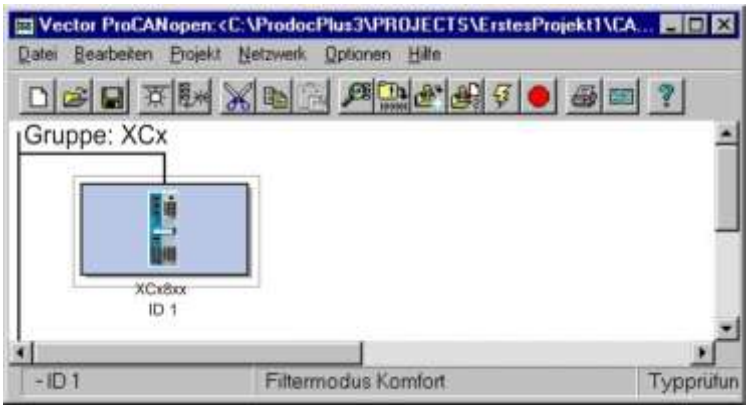
Die aktive CAN-Karte der XCx anwählen. Im Beispiel wird nur die eine standardmäßig vorhandene Karte benutzt. Als Knotennummer (NodeID) kann 1 beibehalten werden. Zum Start von ProCANopen auf den Button *Konfigurieren* klicken (→ Bild 63).

Bild 63: Auswahl der CAN-Karte und Starten von ProCANopen



ProCANopen startet direkt mit dem richtigen CANopen-Projekt (→ Bild 64).

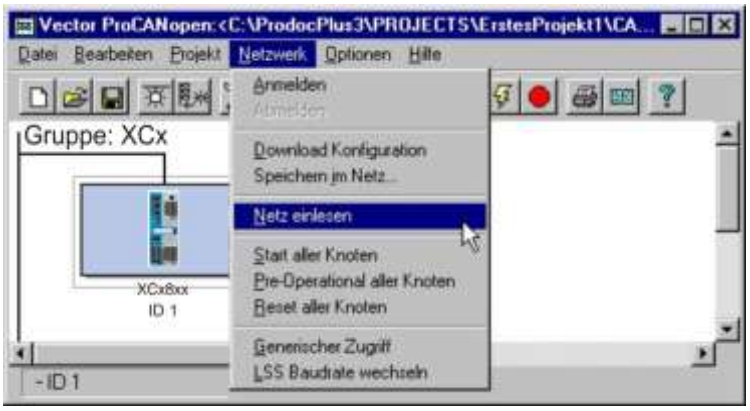
Bild 64: ProCANopen mit aktuellem CANopen-Projekt



6.10.7.3 Erste Verbindungen mit ProCANopen

Zuerst muss das Netzwerk eingelesen werden (→ Bild 65).

Bild 65:
Einlesen des Netzes mit ProCANopen

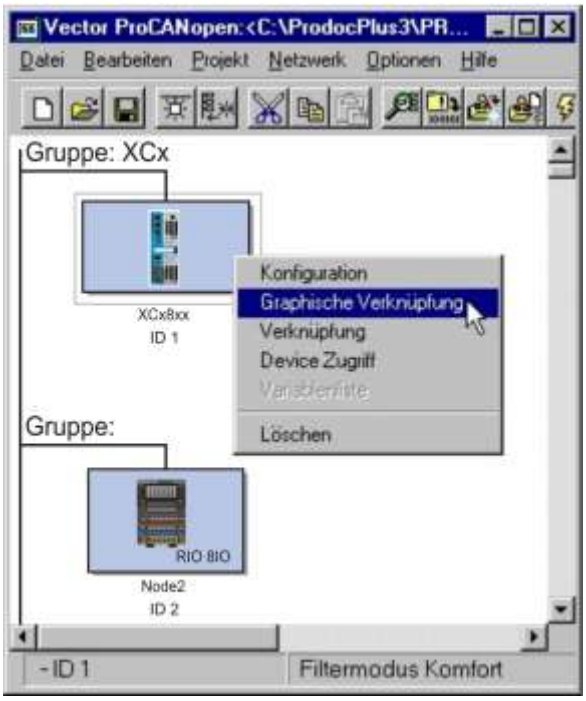


Da das Netzwerk mit "Knoten 1 XCA" bereits vorkonfiguriert ist, muss das Einlesen mit der Scan-Option *Nur Suche neuer Geräte* ausgeführt werden (→ Bild 66).

Bild 66:
Einstellen der Scan-Optionen

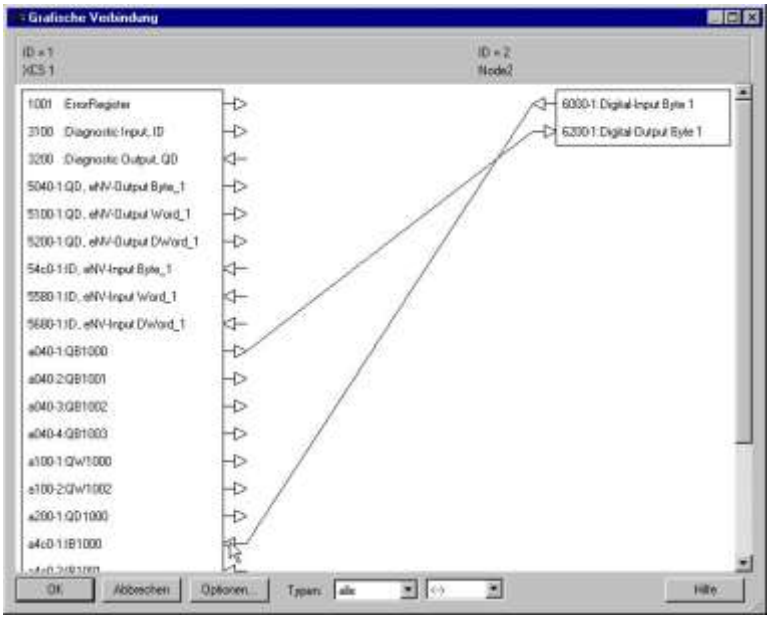
Dann können die Verknüpfungen der Netzwerk-Knoten projektiert werden. Dazu mit der rechten Maustaste auf die XCx klicken, im Kontextmenü des Knotens *Graphische Verknüpfung* auswählen und dann auf den Knoten klicken, mit dem die Verbindung hergestellt werden soll (im Beispiel Knoten 2 RIO 8 I/O) (→ Bild 67).

Bild 67: Graphische Verknüpfung der Netzwerk-Knoten



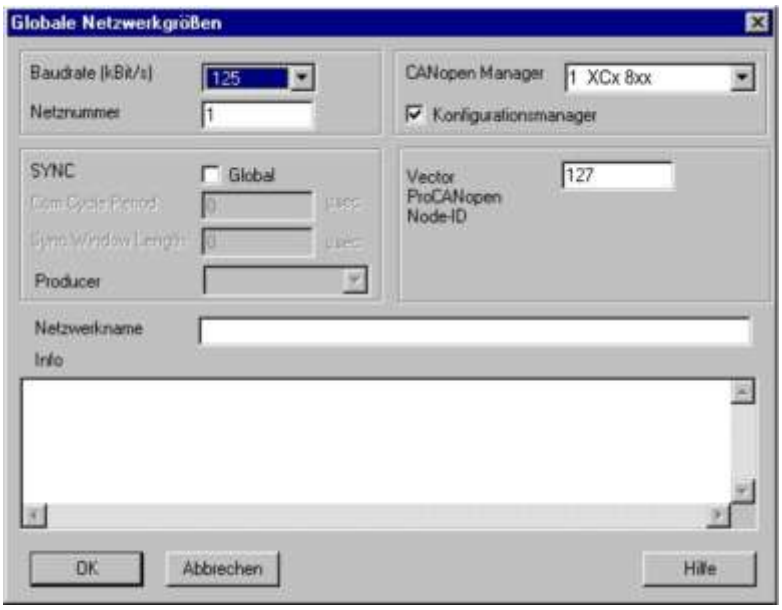
Das nächste Bild zeigt beispielhaft die Kommunikationsbeziehungen zwischen der Steuerung und dem I/O-Modul (→ Bild 68).

Bild 68: Kommunikations beziehung zwischen Steuerung und I/O-Modul



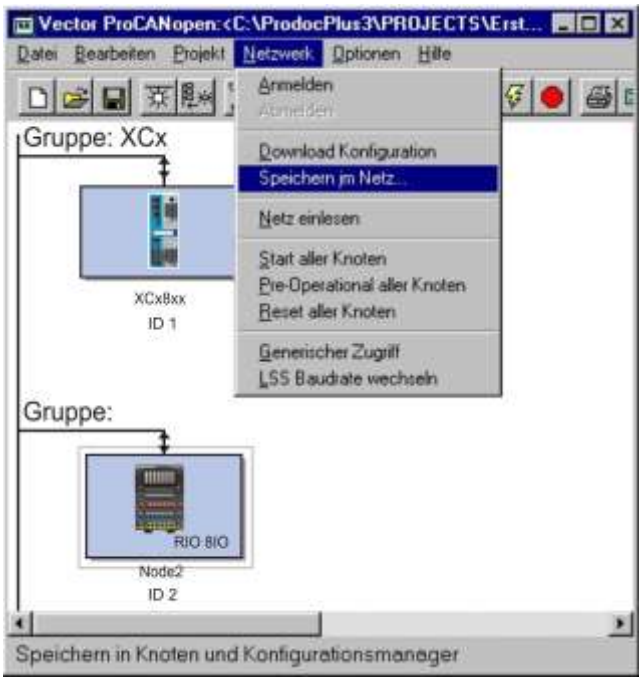
In *Projekt / Globale Konfiguration* den CANopen-Manager "XCx 800" als Konfigurationsmanager auswählen (→ Bild 69).

Bild 69: Auswahl des Konfigurationsmanagers



Mit *Speichern im Netz* wird die CAN-Konfiguration in den Konfigurationsmanager gespeichert (→ Bild 70). Die XCx speichert die Daten in den Compact Flash, und nach dem Einschalten wird das Netzwerk gebootet.

Bild 70: Speichern der CAN-Konfiguration in den Konfigurationsmanager

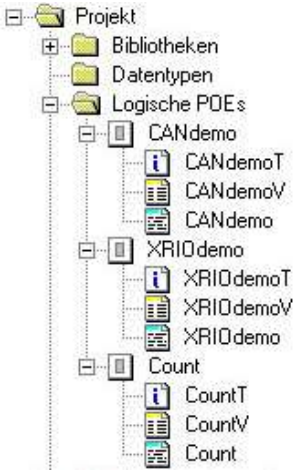


Hinweis

Weitere Hinweise zur CAN-Konfiguration finden Sie in der Betriebsanleitung "Inbetriebnahmehinweise für Feldbussysteme" (→ 10).

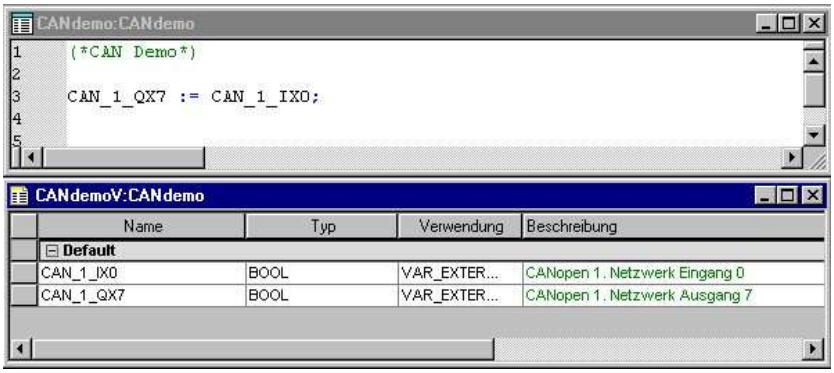
Zum Test der Netzwerkverbindung muss eine neue POE (hier *CANdemo*) erstellt und in der *CanTsk* instanziiert werden (→ Bild 71).

Bild 71:
Test der Netzwerk-verbindung mit POE "CANdemo"



Die POE *CANdemo* mit dem dazugehörigen Variablen-Arbeitsblatt (→ Bild 72):

Bild 72:
POE CANdemo mit dazugehörigem Variablen-Arbeitsblatt



Wird an den Eingang 0 vom RIO 8 I/O CANopen 24 V angelegt, wird der Ausgang 7 auf 1 gesetzt.

6.10.8 Konfiguration der sercos III I/O-Geräte

Die Beschreibung zur Konfiguration der sercos III-Schnittstellen ist ausführlich in der gesonderten Bedienungsanleitung "sercos III-I/O" (R6.322.0770.0) beschrieben.

Diese Betriebsanleitunge kann als PDF-Datei kostenlos von der Website *http://www.schleicher-electronic.com* geladen werden

6.11 Der Webserver

6.11.1 Allgemeine Funktionen und Konzept

- Der Hauptvorteil der Webserver-Technologie ist die Ablage der kompletten Visualisierungs-Applikation (HTML, Java-Applet) auf der Steuerung.
- Die Visualisierungs-Kommunikation zwischen der Steuerung und dem Bediengerät bzw. PC erfolgt über die Standard-Schnittstelle Ethernet mit dem TCP/IP-Protokoll.
- Der Webserver ist im Betriebssystem der XCx integriert und wird über den Schleicher Dialog aktiviert.
- Zum Anzeigen der Visualisierung ist ein Browser auf dem Bediengerät bzw. PC notwendig.

6.11.2 Aktivierung des Webservers

Standardmäßig ist der Webserver auf der XCx inaktiv.

Wird der Webserver für eine Visualisierung benötigt, so muss er über den Schleicher Dialog aktiviert werden über die Dialogseite Inbetriebnahme -> Boot-Einstellungen -> Webserver.

Die steuerungsseitige Default-Einstellung des Root-Verzeichnisses für den Webserver ist:

/pc/webs/web bei der XCx 1100

/ata0/webs/web bei allen anderen XCx Steuerungen

In diesem Verzeichnis wird die Web-Visualisierung abgelegt und der Browser greift hier auf die Web-Seiten zu.

Die Startseite ist die Datei, auf die der Browser zugreift, wenn kein Dateiname im Aufruf der Web-Seite steht (Web-Adresse ohne Dateiname).

6.11.3 Visualisierungsapplikation

Normalerweise ist die Web-Technologie ein Herunterladen in einer Richtung zum Browser, die Web-Seite selbst ist dynamisch (Animation GIF- oder Flash Files). Ein zyklischer Parameterrefresh ist nicht möglich.

Mit Hilfe des Projektierungstools Schleicher MicroVisIT (SpiderControl PLC Editor) kann eine Visualisierung erstellt werden, die anschließend in die Steuerung übertragen wird.

Die Visualisierungs-Applikation besteht aus bildbeschreibenden Dateien und einem Java-Applet, das einen bidirektionalen Datenaustausch zwischen Browser und Steuerung ermöglicht. Dadurch ist ein zyklischer Variablenrefresh, sowie ein Beschreiben von Variablen in der Steuerung, möglich.

6.11.4 Deklaration von Variablen zur Visualisierung

Die Variablen, die visualisiert werden sollen, sind in MULTIPROG mit der Checkbox *PDD* zu markieren (PDD = **P**rocess **D**ata **D**irectory) (→).

Bild 73: Deklaration von PDD-Variablen in MULTIPROG

Darüber hinaus muss in den Einstellungen der Ressource die Einstellung "PDD" -> "Markierte Variablen" eingeschaltet sein.

Diese Variablen werden in der XCx in einer internen Liste aktualisiert. Der Webserver kann die Variablen dieser Liste lesen und schreiben.

Im Projektier-Tool kann dann einem Visualisierungs-Objekt eine Steuerungs-Variable zugeordnet werden. (z.B. @GV.PLC_SYS_TICK_CNT oder @GV.cmpS_lRevisionNo)

6.11.5 Browser / Komponenten

Die in der Steuerung befindliche Visualisierung kann von einem Bediengerät oder von einem PC mit Ethernet aufgerufen werden.

Zur Anzeige der Visualisierung benötigt man den MicroBrowser oder einen Standard-Browser mit installiertem Java.

Ein Aufruf der Visualisierung im Standard-Browser könnte z.B. wie folgt aussehen:

http://192.168.244.2/index.html

Ist die Datei index.html in der Steuerung auch als Start-Datei deklariert, so reicht auch folgender beispielhafter Aufruf aus:

http://192.168.244.2

Die beste visuelle Übereinstimmung der projektierten Seiten erfolgt mit dem MicroBrowser.

6.12 Allgemeine Hinweise zur Inbetriebnahme

Grundinitialisierung

Um bei fehlerhaften SPS-Projekten, Q-Parametern bzw. ungültigen Daten im remanenten Datenspeicher (z.B. bei Ausfall der Pufferbatterie) den Betrieb des Echtzeitbetriebssystems sicherzustellen, kann die Steuerung mit Hilfe der sogenannten Grundinitialisierung in einem abgesicherten Modus starten. Die Grundinitialisierung bewirkt ein Zurücksetzen des SRAMs.

Hiermit besteht die Möglichkeit einer Diagnose der Steuerungsdaten.

Durchführen der Grundinitialisierung

Bitte beachten: Für alle Übergänge gilt jeweils ein Timeout von 4 s.

- Betriebsartenschalter in Position "0".
- Einschalten der Steuerung XCx 800.
- Das gleichzeitige, gelbe Blinken einer LED-Gruppe signalisiert den Start der "GrundInit"-Erkennung.
- Betriebsartenschalter in Position "9", jetzt blinkt nur die LED *RUN/ERR* rot weiter.
- Betriebsartenschalter in Position "0" zurückschalten, die LED-Gruppe blinkt jetzt rot und zeigt die Durchführung der Grundinitialisierung an.

Nach der Grundinitialisierung wird die Steuerung gestartet. Ein vorhandenes SPS-Bootprojekt ist nicht geladen und die SPS startet nicht.

Hinweis

Wird der Drehschalter nach dem Einschalten der Steuerung in Stellung "0" belassen, wird das SPS-Bootprojekt nicht geladen und die SPS startet nicht.

7 Bedienung der XCx mit dem Schleicher Dialog

7.1 Übersicht

Der Schleicher Dialog stellt Bildschirmseiten zur Inbetriebnahme und Bedienung der XCS und XCN zur Verfügung.

Die Installation vom Schleicher Dialog ist beschrieben im Kapitel Schleicher Dialog installieren auf Seite 34.

Je nach verwendeter Steuerung wird zwischen Schleicher Dialog für XCS und XCN unterschieden. Die Erkennung des Steuerungstyps führt das Programm automatisch aus.

Voraussetzung ist ein auf dem PC installierter OPC-Server (siehe -> Konfiguration des OPC-Servers auf Seite96).

Die Installation vom OPC-Server ist beschrieben im Kapitel OPC-Server installieren auf Seite 32.

Die Ethernet-Verbindung des PC mit der XCx muss in Betrieb sein (siehe -> Kommunikationsschema auf Seite 35).

7.2 Vorbereitung des OPC-Servers

7.2.1 Deklaration der OPC-Variablen im SPS-Projekt mit MULTIPROG

Alle Variablen, die der OPC-Server zur Verfügung stellen soll, müssen zuvor in der Variablen-Tabelle des zugehörigen MULTIPROG Projektes als OPC-Variablen gekennzeichnet werden, indem die CheckBox *OPC* für diese Variablen aktiviert wird.

Der Anwender muss die globalen Variablen und / oder die lokalen Variablen einer POE als OPC-Variablen deklarieren.

Die Variablen des Koppelspeichers sind standardmäßig als OPC-Variable deklariert.

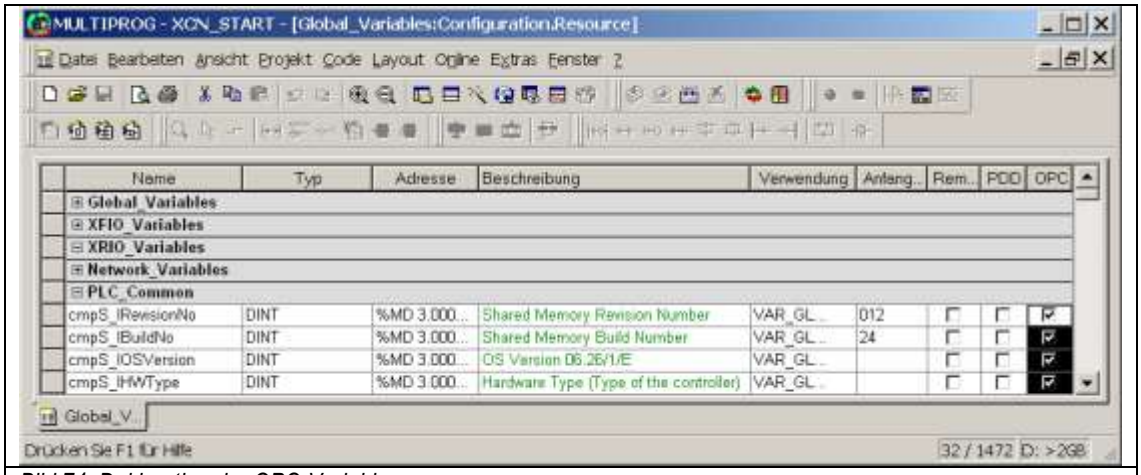


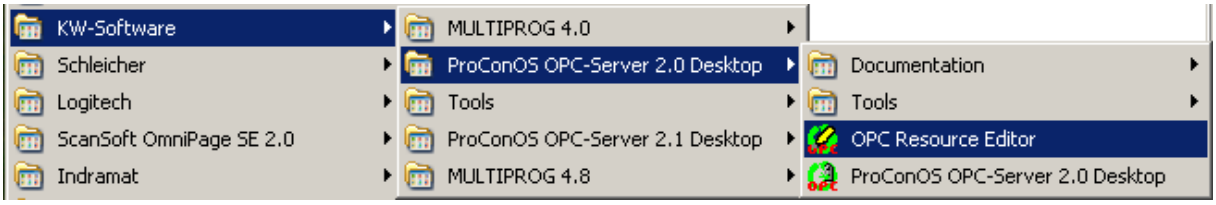
Bild 74: Deklaration der OPC-Variablen

- Übersetzen des SPS-Projektes.
- Übertragen des Projektes mit MULTIPROG an die Steuerung.
- Starten der Steuerung.

7.2.2 Konfiguration des OPC-Servers

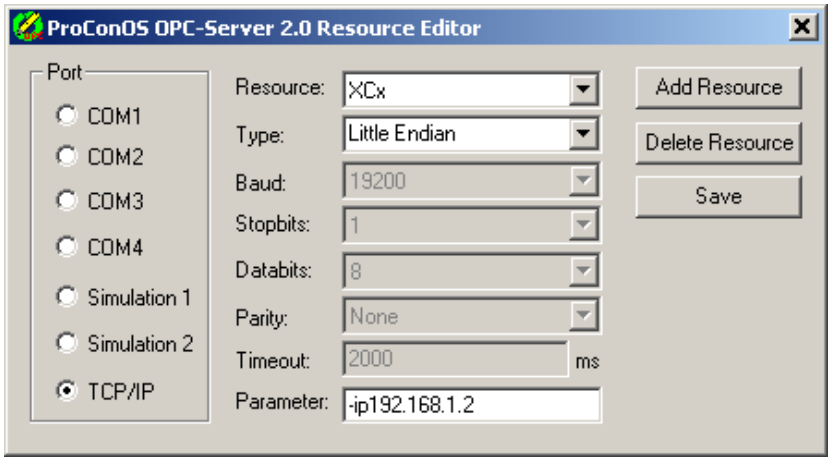
OPC-Server 2.0

OPC-Ressource-Editor starten.
(Über *Start/Programme/...)*



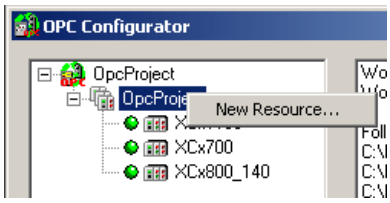
Den Namen der Ressource und die IP-Adresse der Steuerung eingeben. Dazu muss eine neue Ressource angelegt werden (Button *Add Ressource*).

Die neuen Einstellungen abspeichern (Button *Save*) und den OPC-Ressource-Editor beenden (Button *X*).

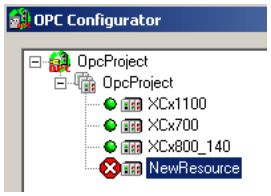


OPC-Server 2.1

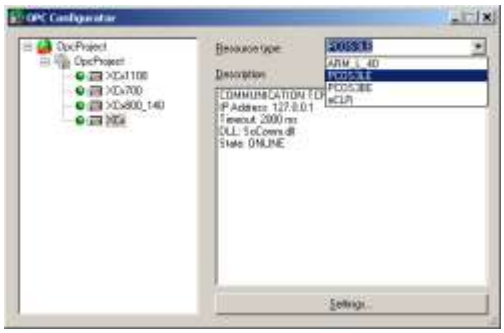
Mit rechter Maustaste auf OpcProject klicken, dann den Button *New Ressource* anklicken.



Das Feld NewResource umbenennen (z.B. XCx).



Im Feld Resource type PCOS3LE auswählen.

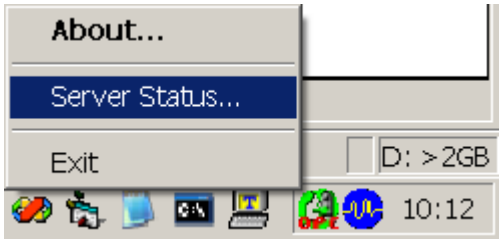


Mit dem Button *Settings* die Eingabemaske *Settings* öffenen, die IP-Adresse eintragen, mit OK bestätigen und den OPC-Editor schliessen.

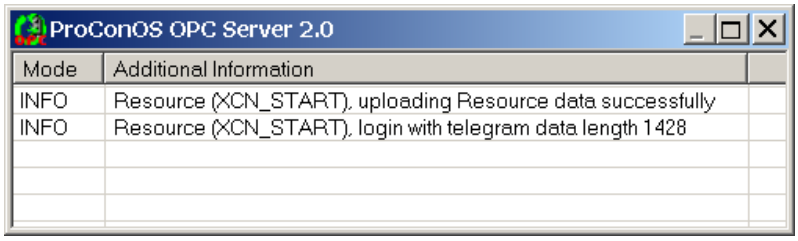
7.2.3 Test des OPC-Servers

ProConOS OPC-Server starten.
(Über *Start/Programme/KW-Software\ProConOS OPC-Server 2.0 Desktop/ProConOS OPC-Server 2.0 Desktop* das Programm starten.)
Es erscheint ein kleines Icon im Windows-System-Tray.

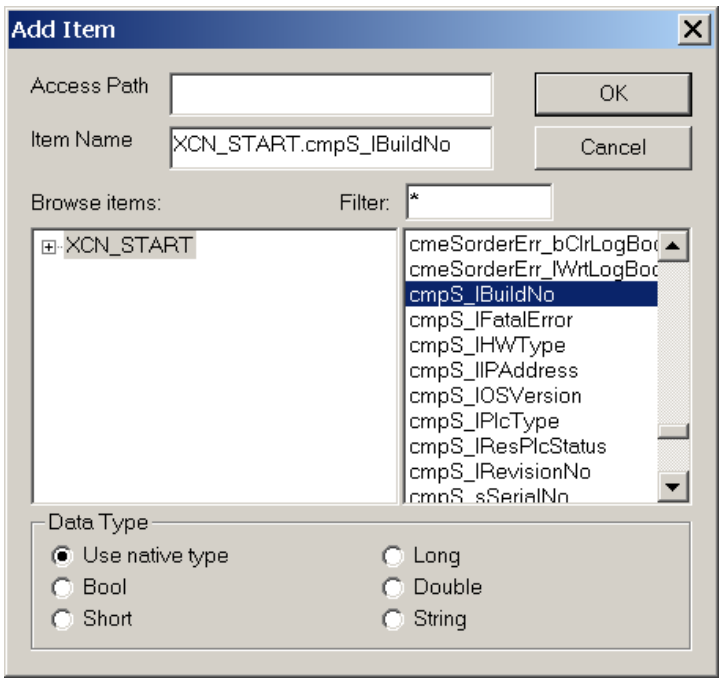
Mit der rechten Maustaste auf dieses Icon klicken und *Server Status*... auswählen.



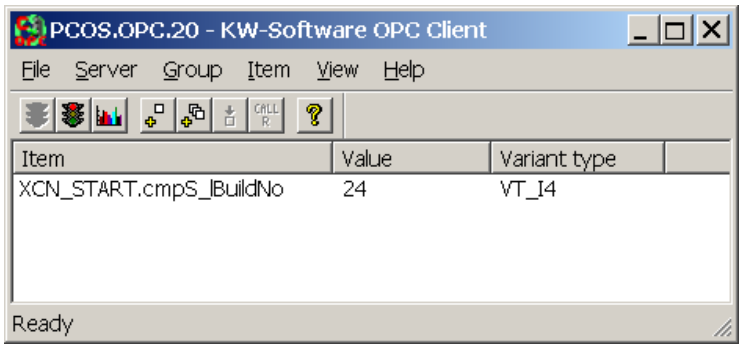
Die Verbindung zur Steuerung wurde ordnungsgemäß erstellt, wenn folgende Meldungen im Statusfenster des OPC-Servers erscheinen.



OPC-Test-Client starten.
(Über *Start/Programme/KW-Software\ProConOS OPC-Server 2.0 Desktop/Tools/OPC Test-Client* das Programm starten.)
Mit dem Befehl *Connect*... im Menü *Server* eine Verbindung zum OPC-Server herstellen.
Mit dem Befehl *Add item* ... im Menü *Group* eine Variable auswählen.



Nach Bestätigung mit OK wird der Wert der ausgewählten Variablen im Hauptfenster des OPC-Test-Clients angezeigt.



7.3 Schleicher Dialog

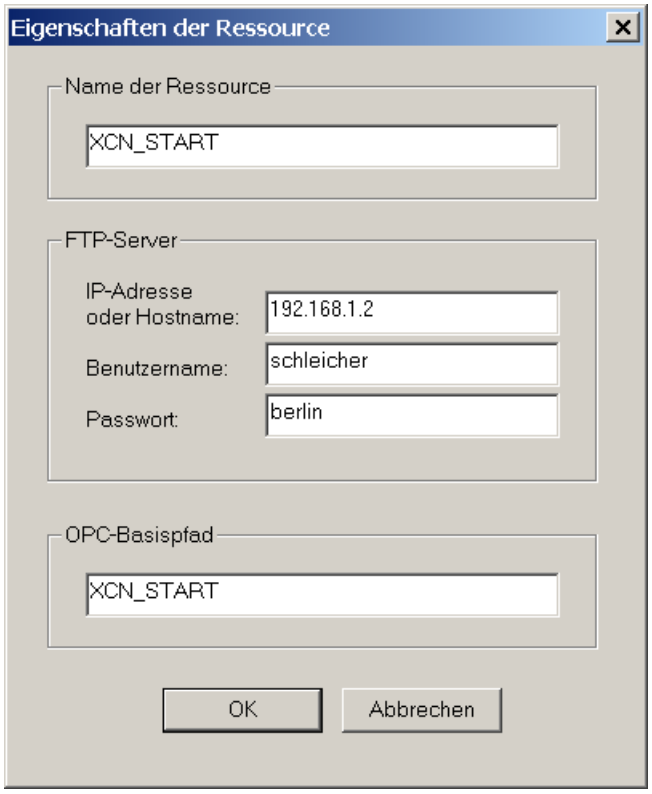
Der Schleicher Dialog stellt alle Dialoge zum Einrichten und Betrieb der CNC und SPS zur Verfügung. Der Schleicher Dialog muss auf dem Service-PC installiert sein.

7.3.1 Schleicher Dialog Konfigurieren

Schleicher Dialog starten.
Nachdem nun alle Einstellungen vorgenommen worden sind, kann der Schleicher Dialog mit PCNC.exe gestartet werden. Beim Start erscheint das Auswahlfenster der Ressource:

Beim ersten Start bzw. weiteres Projekt Hinzufügen:
Eigenschaften anwählen
Die Projektadressen korrekt eingeben z.B.

Name der Ressource:	XCN_START
IP Adresse der Steuerung :	192.168.1.2
Benutzername:	schleicher
Passwort:	berlin



Mit OK können Sie den Schleicher Dialog Starten

7.3.2 Aufbau der Bedienoberfläche

Bild 75: Schleicher-Dialog, Startfenster

Tabelle 45: Schleicher-Dialog, Aufteilung der Bedienoberfläche

Bereich	Bedeutung
1	Status- und Meldebereich
2	Arbeitsbereich für Einstellungen und Informationen
3	Hinweisbereich
4	Softkeys mit Funktionshinweis

Bild 76: Schleicher-Dialog, Status- und Meldebereich

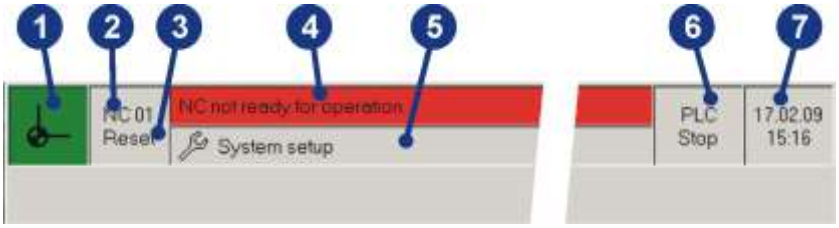


Tabelle 46: Schleicher-Dialog, Status- und Meldebereich

Bereich	Bedeutung
1	Aktuelle Betriebsart anzeigen
2	Angewähltes NC-Teilsystem
3	Aktueller NC-Zustand
4	Meldungfenster
5	Aktuelle Position im Steuerungsmenü
6	SPS-Status
7	Datum und Uhrzeit

7.3.3 Schleicher Dialog SPS/CNC

Steuerungsmenü und Bedienbereiche der XCx

Die oberste Ebene des Steuerungsmenüs besteht aus Bedienbereichen, die sich an den für die Maschine wichtigen Tätigkeiten orientieren (Handbetrieb, Automatik, Programmieren, etc.). Sie werden mit der Tastenkombination <Strg + Funktionstaste> aufgerufen.

Über die untergeordnete Softkey-Ebene (Funktionstasten F1..F8) werden zugehörigen Optionen aufgerufen.

Der Softkey *F1* wird immer zum Aufruf der Hilfeseiten verwendet. Die Hilfeseiten enthalten weitergehende Informationen zum Inhalt der tieferliegenden Bedienebenen.

Handbetrieb <Strg+F1>

F1 Hilfe	F2 Referenzieren	F3 Verfahren	F4	F5	F6	F7 Teilsystem	F8
	Achsen	Jog					
		Schrittmaß					
		Zielwert-vorgabe					
		Handrad					
		Achse					
		Eilgang					
		Nullsetzen					
		Override					

Automatik <Strg+F2>

F1 Hilfe	F2 Programm	F3	F4	F5	F6	F7 Teilsystem	F8
	Aktivieren						
	MDI						
	Einzelsatz						
	Blocksatz						
	Satzfolge						
	Eilgang						
	Override						

Programmieren <Strg+F3>							
F1 Hilfe	**F2 CNC-Programme**	**F3 R-Parameter**	**F4 Werkzeug-daten**	**F5 Nullpunkt-verschie-bungen**	**F6 Koordinaten-systeme**	**F7**	**F8**
	Programm editieren	Wert ändern	Wert ändern	Wert ändern	Wert ändern		
	Neues Programm	Editieren					
	Programm aktivieren	Neu					
	Programm kopieren	Löschen					
	Programm löschen						
	Programm Eigen-schaften						
	Neues Projekt						
	Projekt aktivieren						
	Projekt kopieren						
	Projekt löschen						
	Projekt Eigen-schaften						
	Ansicht						
	Editieren						
	Neu						
	Übertragen						
	Aktua-lisieren						
	Löschen						
	Verzeichnis						

Zugangsberechtigung <Strg+F4> (gilt auch für XCS)							
F1 Hilfe	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**

Fremdsoftware starten <Strg+F5> (gilt auch für XCS)

F1 Hilfe	F2	F3	F4	F5	F6	F7	F8
	Paint	Taschen-rechner	Editor				

Inbetriebnahme <Strg+F6> (Softkeyebene 1)

				(Gilt auch für XCS)			
F1 Hilfe	**F2 CNC-Grundein-stellungen**	**F3 CNC-System**	**F4 Antriebs-konfiguration**	**F5 Boot-Ein-stellungen**	**F6 OPC-Variablen**	**F7 Daten-sicherung**	**F8 Weiter >>**
	Editieren	Editieren	Editieren	Editieren	Wert ändern	Backup	
	Datei-namen	Anzeige-modus	Antriebs-parameter	CAN-Ein-stellungen	Editieren	Löschen	
		Achs-zuordnung	DriveTop		Neu	Öffnen	
					Löschen	Restore	

Inbetriebnahme <Strg+F6> (Softkeyebene 2)

(Gilt auch für XCS)				(Gilt auch für XCS)			
F1 Hilfe	**F2 Programm-einstellungen**	**F3 SPS/CNC-Optionen**	**F4 Protokol-lierung**	**F5 Software Update**	**F6 Verbinden**	**F7 Info**	**F8 << Zurück**
	Eigen-schaften	Optionen freigeben	System-parameter hinzufügen	Update USP			
	Ressource hinzufügen		System-parameter löschen	Update XSF/XSL			
	Ressource löschen		Antriebs-parameter hinzufügen				
	Ressource nach oben		Antriebs-parameter löschen				
	Ressource nach unten						

7.3.4 Aufruf von Activ-Error-Buffer und Log-Book

Die Fehlermeldungen im Active-Error-Buffer und Log-Book sind auf jeder Bedienebene über die Tastenkombination <Strg+?> aufrufbar.

Fehler <Strg+?>							
F1 **Hilfe**	**F2** **Logbuch**	**F3** **Boot-Protokoll**	**F4** **Problem-report**	**F5**	**F6**	**F7**	**F8**
	Öffnen						
	Speichern						
	Drucken						
	Aktuali-sieren						
	Löschen						

8 Die SPS

- Betriebssytem: ProConOS
- Programmierung: MULTIPROG nach IEC 61131-3
- Kommunikation mit der CNC über Koppelspeicher

8.1.1 Programmierung

Die Programmierung der XCx erfolgt mit der Programmiersoftware MULTIPROG nach IEC 61131-3 auf einem PC.

Wichtig!

Die Programmiersoftware besteht aus der Software MULTIPROG und den Add-Ons für MULTIPROG von Schleicher.

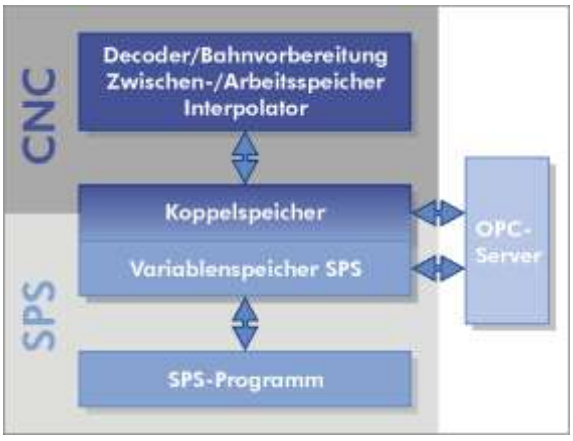
Das Programmiersystem mit Programmieranleitung ist als Zubehör zu beziehen (→ Seite 134).

Die SPS wird mit dem fertig konfigurierten Projekt ausgeliefert, auf dessen Grundlage die Programmierung der SPS begonnen werden kann (→ " Erste Schritte mit MULTIPROG", Seite 41).

8.1.2 Der Koppelspeicher

Der Koppelspeicher der XCN verbindet die Ablaufsteuerung der SPS und die Bewegungsfunktionen der CNC. Die beiden Steuerungssysteme arbeiten zum Datenaustausch synchron auf den Speicher, wobei die SPS eine Masterfunktion übernehmen kann. Auch Visualisierungssysteme sind über OPC in die Kommunikation eingebunden.

Bild 77: Koppelspeicher als Verbindung von SPS und CNC



Die enge Verbindung des SPS- mit dem CNC-System ermöglicht die Realisierung von komplexen Funktionsabläufen, die mit getrennten SPS- und CNC-Steuerungen undenkbar wären. Die klassische SPS-Schnittstelle ermöglicht SPS-Funktionen im NC-Programm, z.B. das Setzen und Abfragen von SPS-Merkern. Die Synchronisierung der SPS-Task mit der CNC-Lageregelung bietet darüber hinaus weitreichende Optionen:

- Es entstehen keine Wartezeiten und kein Kommunikations-Overhead
- Die SPS kann alle Aktionen der CNC überwachen
- Die CNC kann über die SPS Sollwerte ausgeben
- Die SPS kann der CNC Führungsgrößen im Lageregeltakt vorgeben

Die Inbetriebnahme der CNC-Funktionen werden in der Betriebsanleitung CNC-Inbetriebnahme beschrieben (→ Seite 10).

8.2 SPS-Betriebszustände und Startverhalten

8.2.1 Betriebszustände

Betriebszustand	Beschreibung
EIN	• Es ist kein Programm geladen
STOPP	• Programm ist geladen • Anwendertasks sind inaktiv • Eingänge des Prozessabbildspeichers werden nicht aktualisiert • Ausgangssignale werden nicht an die Ein- und Ausgänge übermittelt
BETRIEB	• Programmausführung ist aktiviert • Anwendertasks sind aktiv • Eingänge des Prozessabbildspeichers werden gemäß der I/O-Konfiguration aktualisiert • Ausgänge des Prozessabbildspeichers werden gemäß der I/O-Konfiguration und der Programmausführung aktualisiert
HALT	• Programmausführung wird an einem Haltepunkt angehalten • Anwendertasks sind inaktiv • Eingänge des Prozessabbildspeichers werden nicht aktualisiert • Ausgänge des Prozessabbildspeichers werden nicht aktualisiert

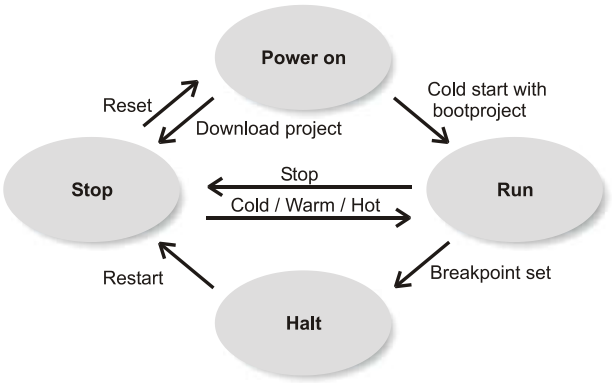
Tabelle 47: SPS-Betriebszustände

Der aktuelle Zustand der SPS wird in MULTIPROG im Projekt-Kontrolldialog in der Zeile *Status* angezeigt. Wenn hinter dem aktuellen Zustand im Kontrolldialog 'Debug' angezeigt wird, bedeutet das, dass Haltepunkte gesetzt oder Variablen geforct wurden.

8.2.2 Wechseln der Betriebszustände mit MULTIPROG

Über die graphische Benutzeroberfläche von MULTIPROG kann gesteuert werden, wann die Programmausführung auf der SPS gestartet und gestoppt wird. Die Schaltflächen für Wechsel, die im aktuellen Betriebszustand nicht möglich sind, sind im Projekt-Kontrolldialog abgeblendet (→ Bild 78).

Bild 78: MULTIPROG, Wechseln der Betriebszustände



Starten der Programmausführung

Zustandswechsel von → nach	Schaltfläche im Kontrolldialog	Beschreibung, was passiert
Stopp → Betrieb	Cold	• es erfolgt ein Kaltstart • alle Daten werden initialisiert • SPG 1 wird aufgerufen • alle Anwendertasks werden aktiviert • die Programmausführung wird aktiviert
Stopp → Betrieb	Warm	• es erfolgt ein Warmstart • nur nicht-gepufferte Daten werden initialisiert • SPG 0 wird aufgerufen • alle Anwendertasks werden aktiviert • die Programmausführung wird aktiviert
Stopp → Betrieb	Hot	• es erfolgt ein Heißstart • es werden keine Daten initialisiert • alle Anwendertasks werden aktiviert • die Programmausführung wird aktiviert • nicht verfügbar, wenn Sie die Programmausführung zum ersten Mal nach dem Senden starten

Tabelle 48: MULTIPROG, Starten der Programmausführung

Stoppen der Programmausführung

Zustandswechsel von → nach	Schaltfläche im Kontrolldialog	Beschreibung, was passiert
Betrieb → Stopp		• alle Anwendertasks werden deaktiviert, wenn ihr Arbeitszyklus beendet ist • SPG 2 wird aufgerufen • die Ausgänge des Prozessabbildspeichers werden geschrieben • die Programmausführung wird gestoppt • die physikalischen Ausgänge werden auf Null oder Vorzugsabschaltlage gesetzt

Tabelle 49: MULTIPROG, Stoppen der Programmausführung

Allgemeines Reset

Zustandswechsel von → nach	Schaltfläche im Kontrolldialog	Beschreibung, was passiert
Stopp → Ein		• das Projekt wird gelöscht • es erfolgt ein allgemeines Reset

Tabelle 50: MULTIPROG, Allgemeines Reset

8.3 Systemvariablen

Systemvariablen informieren über den Systemzustand, wie z.B. über geforcte Variablen, Leistungsfähigkeit der CPU, etc. Diese Variablen haben feste Speicheradressen und können vom SPS-Programm verwendet werden, um die entsprechenden Informationen zu erhalten.

Alle Systemvariablen in der folgenden Tabelle sind bereits im Bereich *Global_Variables* des Arbeitsblattes *Global_Variables* deklariert.

Name	Datentyp	Log. Adr. (Byte)	Log. Adr. (Bit)	Beschreibung
PLCMODE_ON	BOOL	0	0	TRUE := aktueller SPS-Zustand ist EIN
PLCMODE_RUN	BOOL	0	1	TRUE := aktueller SPS-Zustand ist BETRIEB
PLCMODE_STOP	BOOL	0	2	TRUE := aktueller SPS-Zustand ist STOPP
PLCMODE_HALT	BOOL	0	3	TRUE := aktueller SPS-Zustand ist HALT
PLCDEBUG_BPSET	BOOL	1	4	TRUE := ein oder mehrere Haltepunkte sind gesetzt
PLCDEBUG_FORCE	BOOL	2	0	TRUE := eine oder mehrere Variablen sind geforct
PLCDEBUG_POWERFLOW	BOOL	2	3	TRUE := Durchlaufkontrolle ist aktiv
PLC_TICKS_PER_SEC	INT	44	-	Anzahl der Systemticks pro Sekunde, die von der SPS als Systemzeitbasis verwendet werden. Dieser Wert bestimmt die Zeitauflösung der SPS für Funktionsbausteine für Zeitverzögerung, wie TON, TOF oder TP und die kürzeste Zykluszeit für die DEFAULT-Task und zyklische Tasks.
PLC_SYS_TICK_CNT	DINT	52	-	Anzahl der gezählten SPS Systemticks

Tabelle 51: Systemvariablen

Zusätzlich zu diesen Systemvariablen sind weitere Variablen definiert, die Informationen zum System vorhalten. Die Typdefinitionen der Variablen sind in der Bibliothek "SchleicherLib" im Abschnitt *PLC_Types* zu finden.

8.4 Bibliotheken und Funktionsbausteine in MULTIPROG

Funktionsbausteine sind in Bibliotheken zusammengefasst. Sie werden, je nach Steuerungstyp, beim Erstellen eines neuen MULTIPROG-Projektes automatisch eingebunden oder können bei Bedarf manuell eingebunden werden.

Wichtig!

Grundsätzlich muss die zum Steuerungsbetriebssystem passende, aktuellste Version der Bibliotheken eingesetzt werden.

Bibliotheken	XCA 11xx	XCN 8xx	XCS 8xx	XCN 7xx	XCS 7xx	XCN 5xx	XCS 5xx	XCN 4xx	XCS 4xx	XCN 3xx	XCS 3xx	XCS 20C	XCS 20P	ProNumeric	ProSyCon	MCS 20-20	MCS 20-21	Simulation
PROCONOS	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
BIT_UTIL	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
CANopen_Vxxx	+	+	+	+	+	+	+	+	+	o	o	+	–	+	+	–	+	–
CFB_Vxxx	o	o	o	o	o	o	o	o	o	o	o	–	–	o	o	–	–	–
CNC_Vxxx	+	+	–	+	–	+	–	+	–	+	–	–	–	+	–	–	–	–
Date_Time	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	–
MC_Vxxx	–	–	–	o	–	o	–	–	–	o	–	–	–	–	–	–	–	–
Microline	–	–	–	–	–	–	–	–	–	–	–	+	+	–	–	+	+	–
MMI	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	–
PLC_Vxxx	+	+	+	+	+	+	+	+	+	+	+	o	o	+	+	–	–	–
PNS_Vxxx	–	–	–	–	–	o	o	–	–	–	–	–	–	–	–	–	–	–
Profibus_Vxxx	o	o	o	o	o	o	o	o	o	o	o	–	–	–	–	–	–	–
SchleicherLib_Vxxx	+	+	+	+	+	+	+	+	+	+	+	–	–	+	+	–	–	–
Serial	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	–
XCx7_Vxxx	+	+	+	+	+	–	–	+	+	–	–	–	–	–	–	–	–	–

\+ Werden beim Erstellen eines neuen Projektes automatisch eingebunden.
o Können je nach Bedarf manuell eingebunden werden.
– Nicht möglich oder unnötig.

Tabelle 52: Bibliotheken und Funktionsbausteine in MULTIPROG

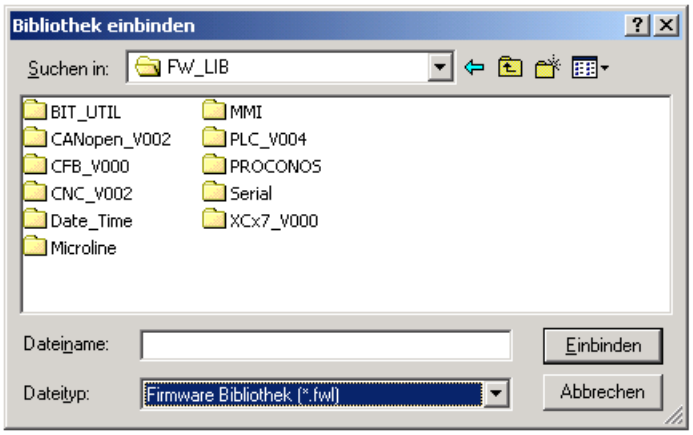
Funktionsbausteine können folgendermaßen eingebunden werden:

- Mit der rechten Maustaste im Projektbaum von MULTIPROG das Kontextmenü *Bibliotheken / Einfügen / Bibliothek* öffnen (→ Bild 79).

Bild 79: MULTIPROG, Kontextmenü "Bibliotheken"

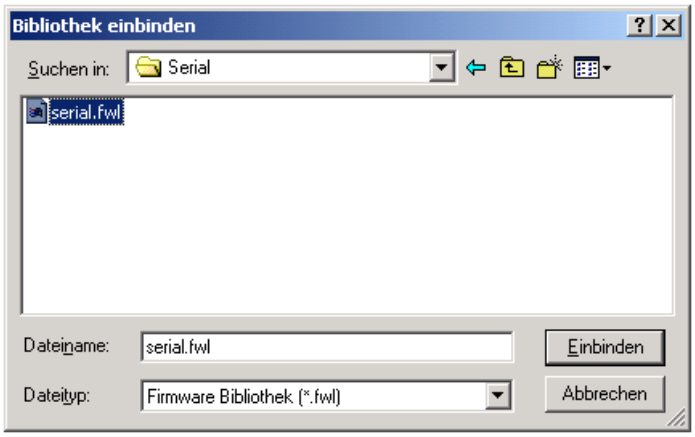
- Den Pfad *.\KWSoft\MWT\PLC\FW_LIB* und den Dateityp *Firmware Bibliothek (*.fwl)* wählen (→ Bild 80).

Bild 80: MULTIPROG, Dialogfenster "Bibliothek einbinden"



Jede Bibliothek ist in einem eigenen Pfad gespeichert. Soll z.B. die Bibliothek *Serial* eingebunden werden, muss sie im gleichnamigen Pfad der Bibliothek ausgewählt werden (→ Bild 81).

Bild 81: MULTIPROG, Einbinden der Bibliothek "Serial"



Zu den Bibliotheken (außer SchleicherLib) ist eine Online-Hilfe vorhanden. Die Online-Hilfe ist über das Kontextmenü der jeweiligen Bibliothek erreichbar. Das Kontextmenü wird aktiv, wenn mit der rechten Maustaste auf das Icon der Bibliothek geklickt wird.

8.4.1 Hinweis zu den Variablendeklarationen der Beispielprogramme von FBs

Die Beispielprogramme in den Hilfen zu den Funktionsbausteinen enthalten Variablendeklarationen nach IEC 61131-3 mit den Schlüsselwörtern VAR und END_VAR. Sollen die Beispielprogramme mit MULTIPROG angewendet werden, müssen die Variablen-deklarationen in Tabellenform auf dem Variablen-Arbeitsblatt der benutzten POE von Hand eingetragen werden.

8.4.2 Bibliothek CANopen_Vxxx

Die Bibliothek "CANopen_Vxxx" enthält Funktionsbausteine für die Parametrierung und Diagnose des CANopen-Netzwerkes.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
CO_NET_SDO_WRITE	150	sendet ein Service Data Object (SDO)	XCx ProNumeric ProSyCon MCS 20-21 XCS 20C
CO_NET_SDO_READ	151	empfängt ein Service Data Object (SDO)	
CO_NET_GET_LOCAL_NODE_ID	152	liefert die eigene Node-ID zurück	
CO_NET_GET_STATE	153	liefert den aktuellen CANopen-Status	
CO_NET_GET_KERNEL_STATUS	154	liefert den sog. erweiterten CANopen-Kernelstatus	
CO_NET_NMT	155	setzt den Status eines oder aller Geräte im CANopen-Netzwerk	
CO_NET_RECV_EMY_DEV	156	liest etwaige Emergency-Nachrichten von einem bestimmten Netzwerk-Knoten	
CO_NET_RECV_EMY	157	liest etwaige Emergency-Nachrichten von einem beliebigen Netzwerk-Knoten	
CO_NET_RECV_ERR_DEV	160	liest etwaige Error-Nachrichten von einem bestimmten Netzwerk-Knoten	
CO_NET_RECV_ERR	161	liest etwaige Error-Nachrichten von einem beliebigen Netzwerk-Knoten	
CO_NET_SENDL2	162	sendet beliebige CAN Layer 2-Nachrichten	
CO_NET_PING	163	führt ein Ping auf einen bestimmten Netzwerk-Knoten aus	
CO_NET_RESTART_CAN	164	startet die CANopen Kommunikation neu (z.B. nach "bus-off")	
CO_NET_RESTART_ALL	165	startet den kompletten CANopen-Stack neu	
CO_NET_SHUTDOWN	166	stoppt den CANopen-Stack	
CO_NET_CAN_SYNC	170	ermöglicht die Synchronisation zwischen SPS-Task und den CANopen-Stack	

Tabelle 53: Bibliothek CANopen_Vxxx

8.4.3 Bibliothek CFB_Vxxx

Die an IEC 61131-5 angelehnte Bibliothek "CFB_Vxxx" enthält Funktionsbausteine zur Peer-to-Peer-Kommunikation über TCP/IP.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
CONNECT_V	60	stellt eine Peer-to-Peer-Verbindung zwischen zwei Teilnehmern her	XCx ProNumeric ProSyCon
USEND_V	61	sendet beliebige Daten	
URCV_V	62	empfängt beliebige Daten	

Tabelle 54: Bibliothek CFB_Vxxx

8.4.4 Bibliothek CNC_Vxxx

Die Bibliothek "CNC_Vxxx" enthält Funktionsbausteine für das Lesen und Schreiben von Systemdaten, SERCOS-, XRIO- und CAN-Antriebsparametern und PROFIBU-DP-Antriebsparametern.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
READ_Q_PARAM_*	200 bis 207	liest einen CNC-Systemdaten-Parameter	XCN ProNumeric
WRITE_Q_PARAM_*	208 bis 215	schreibt einen CNC-Systemdaten-Parameter	
SAVE_Q_PARAM_*	221	speichert die CNC-Systemdaten-Parameter auf der Festplatte	
SAVE_R_PARAM_*	220	speichert die CNC-Rechenparameter auf der Festplatte	
READ_SERC_PARAM	302	liest einen SERCOS-Parameter	
WRITE_SERC_PARAM	303	schreibt einen SERCOS-Parameter	
SET_SERC_PHASE	304	Umschaltung der SERCOS-Kommunikationsphase	
SET_SERC_COMMAND	308	ausführen eines SERCOS-Kommandos	
MC_ANALOG	300	XRIO Motion Control Baustein (mit Lageregler)	XCN
MC_ANALOG_1_AXIS	307	XRIO Motion Control Baustein für eine Achse (mit Lageregler)	
READ_AXIS_PAGE	305	liest einen Parameter aus der einer Achse zugeordneten sogenannten Remotepage.	
WRITE_AXIS_PAGE	306	schreibt einen Parameter in die einer Achse zugeordneten sogenannten Remotepage	
MC_CAN_1_AXIS	315	CAN MotionControl Baustein für eine Achse	XCN * Für XCN700 ab V09.38/0
MC_CAN	301	CAN MotionControl Baustein	
MC_DP	309	PROFIBUS-DP Motion Control Baustein	
MC_DP_1_AXIS	310	PROFIBUS-DP Motion Control Baustein für eine Achse	

Tabelle 55: Bibliothek CNC_Vxxx

Die Funktionsbausteine READ_AXIS_PAGE und WRITE_AXIS_PAGE sind von der Bibliothek XCx7_Vxxx in die Bibliothek CNC_Vxxx übernommen worden (ab CNC_V006 / XCx7_V002).

* Der Funktionsbaustein MC_CAN_1_AXIS ist ab V10.03/0 in CNC_V009.lib verfügbar.

8.4.5 Bibliothek Date_Time

Die XCx verfügt über eine gepufferte Echtzeituhr mit Kalender (Berücksichtigung von Schaltjahren) und einer Auflösung von 1 s.

Datum und Uhrzeit können mit den Funktionsbausteinen aus der Bibliothek "Date_Time" gelesen und gesetzt werden.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
GET_TIME	130	Zeit lesen	XCx ProNumeric ProSyCon MCS xx-xx XCS 20x
GET_DATE	128	Datum lesen	
SET_TIME	131	Zeit setzen	
SET_DATE	129	Datum setzen	

Tabelle 56: Bibliothek Date_Time

8.4.6 Bibliothek MC_Vxxx

Die Bibliothek "MC_Vxxx" (Motion Control) enthält Funktionsbausteine zur Programmierung von Bewegungsabläufen in der SPS.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
MC_MoveAbsolute	320	Achse wird beauftragt, auf eine absolute Position zu fahren	XCN 300 XCN 5xx XCN 700
MC_MoveRelative	321	Achse wird beauftragt, eine Strecke zu fahren	
MC_MoveAdditive	322	Achse wird beauftragt, auf eine absolute Position zu fahren	
MC_MoveVelocity	324	Achse wird beauftragt, mit der vorgegebene Geschwindigkeit zu fahren	
MC_Home	325	Achse wird beauftragt, zu referenzieren	
MC_Stopp	326	Achse wird beauftragt, Achsbewegung zu beenden	
MC_Power	327	Achse wird beauftragt, Drehmoment (Reglerfreigabe) einzuschalten	
MC_ReadStatus	328	Status-Informationen der Achsen werden gelesen	
MC_ReadAxisError	329	Aktuelle Fehlernummer wird gelesen	
MC_Reset	330	Reset (Fehlerquittierung) wird durchgeführt	
MC_ReadParameter	331	Ein Parameter der Achse wird gelesen	
MC_ReadBoolParameter	332	Ein boolscher Parameter der Achse wird gelesen	
MC_WriteParameter	333	Ein Parameter der Achse wird geschrieben	
MC_WriteBoolParameter	334	Ein boolscher Parameter der Achse wird geschrieben	
MC_ReadActualPosition	335	Aktuelle Achsposition wird gelesen	
MC_GetCncAxis	345	Achse wird von der CNC ausgeliehen, um sie in der SPS verfahren zu können	
MC_ReleaseCncAxis	346	Ausgeliehene Achse wird zur CNC zurückgegeben	

Tabelle 57: Bibliothek MC_Vxxx

8.4.7 Bibliothek MMI

Die Bibliothek "MMI" realisiert die Kommunikation mit einem Bediengerät der COP-Familie über die serielle Schnittstelle der Steuerung.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstypen
PPF_COP_COMM	140	kommuniziert mit einem COP-Bediengerät (PNet-Protokoll)	XCx ProNumeric ProSyCon MCS xx-xx XCS 20x

Tabelle 58: Bibliothek MMI

8.4.8 Bibliothek PLC_Vxxx

Über den Umfang der Standard IEC- bzw. ProConOS-Funktionsbausteine hinaus werden weitere, steuerungsspezifische Firmware-Funktionsbausteine in dieser Bibliothek bereitgestellt.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungstypen
PUT_ERROR	400	erzeugt eine nutzerdefinierte Fehlermeldung (bitte nicht mehr verwenden)	XCx ProNumeric ProSyCon
PUT_ERROR2	401	erzeugt eine nutzerdefinierte Fehlermeldung	
CLEAR_ERROR	402	löscht eine mit Lock-Flag abgesetzte Fehlermeldung	
READ_FILE	405	lesender Dateizugriff	
WRITE_FILE	406	schreibender Dateizugriff	
SEND_MAIL	410	sendet eine E-MAIL (SMTP Client)	
XFIO_CONFIG	420	XFIO Interrupt Konfiguration	XCx XCS 20x
XRIO_STATE	422	XRIO Statusinformationen	
GET_MTS	430	liefert den aktuellen Zeitwert in µs-Ticks	XCx ProNumeric ProSyCon XCS 20x
OPEN_PROFILE	431	öffnet eine Datei im INI-Format	
NEW_PROFILE	432	legt eine neue Datei im INI-Format an	
FLUSH_PROFILE	433	schreibt aktualisierte Datei im INI-Format	
CLOSE_PROFILE	434	schließt eine Datei im INI-Format	
GET_PROFILE_STRING	435	liest einen String aus einer Datei im INI-Format	
GET_PROFILE_INT	436	liest einen Integer-Wert aus einer Datei im INI-Format	
GET_PROFILE_REAL	437	liest einen Real-Wert aus einer Datei im INI-Format	
WRITE_PROFILE_STRING	438	schreibt einen String in eine Datei im INI-Format	
WRITE_PROFILE_INT	439	schreibt einen Integer-Wert in eine Datei im INI-Format	
WRITE_PROFILE_REAL	440	schreibt einen Real-Wert in eine Datei im INI-Format	

Tabelle 59: Bibliothek PLC_Vxxx

8.4.9 Bibliothek PNS_Vxxx

Die Bibliothek "PNS_Vxxx" enthält Funktionsbausteine für die Parametrierung und Diagnose des PROFINET-Netzwerkes.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
PNSReadIOData	530	liest IO-Daten	XCx 5xx
PNSWriteIOData	531	schreibt IO-Daten	
PNSCommunicating	532	liefert Status von PROFINET-Verbindung	

Tabelle 60: Bibliothek PNS_Vxxx

8.4.10 Bibliothek Profibus_Vxxx

Die Bibliothek "Profibus_Vxxx" enthält Funktionsbausteine für die Kommunikation über die PROFIBUS-Karte.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
DP_NET_GET_STATE	190	liefert den Status der PROFIBUS-Karte	XCx
DP_NET_PUT_MSG	191	setzt eine Nachricht an das Message-Interface der Hilscher-Karte ab	
DP_NET_GET_MSG	192	holt eine Nachricht vom Message-Interface der Hilscher-Karte ab	

Tabelle 61: Bibliothek Profibus_Vxxx

8.4.11 Bibliothek SchleicherLib_Vxxx

Die Bibliothek "SchleicherLib_Vxxx" enthält Datentypdefinitionen der Firmware, die für MULTIPROG bereitgestellt werden. Funktionsbausteine sind in dieser Bibliothek nicht enthalten.

8.4.12 Bibliothek Serial

Die Bibliothek "Serial" enthält Funktionsbausteine für die serielle Kommunikation der Steuerungen.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
PORT_OPEN	135	öffnet eine serielle Schnittstelle	XCx
PORT_CLOSE	136	schließt eine serielle Schnittstelle	ProNumeric
PORT_READ	137	gibt Zeichen auf einer seriellen Schnittstelle aus	ProSyCon
PORT_WRITE	138	liest Zeichen von einer seriellen Schnittstelle	MCS xx-xx
PORT_STATE	139	liefert Statusinformation einer seriellen Schnittstelle	XCS 20x

Tabelle 62: Bibliothek Serial

8.4.13 Bibliothek XCx7_Vxxx

Die Bibliothek "XCx7_Vxxx" enthält Funktionsbausteine die ausschließlich bei den Steuerungstypen XCx 700, XCx 400, XCx 800 und XCx 1100 verwendet werden.

Funktionsbaustein	Nr.	Kurzbeschreibung	Steuerungs-typen
UZB_VR	250	Funktionsbaustein für den Betrieb der UZB 2VR Module	XCx 4xx XCx 7xx XCx 8xx XCx 11xx
UBA_ERR_CTRL	251	Fehlerhandling der UBA-Erweiterungsmodule	
READ_RP	252	lesender Zugriff auf System-U Remotepages	
WRITE_RP	253	schreibender Zugriff auf System-U Remotepages	
IBSM	254	InterBus-S Master (USK-DIM)	

Tabelle 63: Bibliothek XCx7_Vxxx

9 Das Multi-Task-System

9.1 Übersicht

Basis ist ein Echtzeit-Betriebssystem, das durch Taskprioritäten gesteuert wird. Im Programmiersystem MULTIPROG wird ein Programm einer Task zugewiesen. Den Tasks wiederum werden unterschiedliche Prioritätsstufen und Zeiten zugeordnet, die die Reihenfolge und Dauer der Bearbeitung gemäß ihrer Wichtigkeit sicherstellen.

Drei Prioritätsstufen für Tasks werden verwendet (→ Bild 82):

Bild 82: Multi-Task-System, Prioritätsstufen

- **Überwachungstask**
  Die Überwachungstask ist eine besonders geschützte Betriebssystemtask, die in der höchsten Prioritätsstufe abgearbeitet wird. Sie ermittelt Fehler, wie z.B. eine Division durch Null oder die Überschreitung der Ausführungszeit einer Task, und aktiviert die entsprechende Betriebssystemtask.
- **Anwender- und Defaulttasks**
  Auf der Anwender- und Defaulttask-Stufe laufen alle Tasks, die vom Anwender eingefügt werden. In diesem Bereich laufen auch einige wichtige Firmware-Tasks, die beim Parametrieren der Anwender-Tasks berücksichtigt werden müssen. Siehe dazu Abschnitt "Task-Prioritäten", Seite 125. Die Anwendertasks sind zeitüberwacht (Watchdog).
  - **Zyklische Tasks**
    führen die ihnen zugewiesenen Programme innerhalb eines definierten Zeitintervalls mit einer vom Anwender vorgegebenen Priorität aus. Die Task mit der höchsten Priorität wird als erste aufgerufen.
  - **Ereignistasks**
    werden vom Betriebssystem der Steuerung gestartet, wenn bestimmte Ereignisse wie z.B. Interruptsignal, CANopen- oder IPO-Task auftreten.
  - **Defaulttask**
    ist die Anwendertask mit der niedrigsten Priorität. Sie ist nicht zeitüberwacht und wird als Hintergrundtask aktiviert, wenn zum entsprechenden Zeitpunkt keine höherprioren Anwendertasks aktiv sind.
- **Betriebssystemtasks**
  Auf der Prioritätsstufe für Betriebssystemtasks laufen Tasks z.B. für Kommunikation, Debugging, Speicherverwaltung und Systemkontrolle vom Anwender unbeeinflusst ab.

9.2 Anwender-Task

Achtung!

Eine falsche oder zumindest ungeeignete Wahl der Anwender-Task-Einstellungen hinsichtlich Typ, Priorität oder Interrupt- Mode usw. - insbesondere in Kombination mit langen Programmlaufzeiten - kann zu Steuerungsfehlfunktionen führen, da essentielle Betriebssystem-Tasks verdrängt werden.

Beschreibung im Abschnitt Task-Prioritäten unbedingt beachten.

Anwender-Tasks sind alle Tasks, die durch den Anwendungsprogrammierer eingefügt werden.

Die Default-Task gehört ebenfalls zur Prioritätsstufe für Anwender-Tasks. Sie ist die Anwender-Task mit der niedrigsten Priorität. Die Default-Task wird abgearbeitet, wenn zum entsprechenden Zeitpunkt keine Anwender-Task aktiv ist.

Es können verschiedene Anwender-Tasktypen verwendet werden.

9.2.1 Zyklische Task

Zyklische Tasks führen die ihnen zugewiesenen Programme innerhalb eines definierten Zeitintervalls mit einer vom Anwender vorgegebenen Priorität aus.

In MULTIPROG können den einzelnen Tasks Prioritäten zwischen 0 und 31 zugeordnet werden. 0 steht für die höchste, 31 für die niedrigste Priorität. Die Task mit der höchsten Priorität wird als erste aufgerufen. Die User-Task-Prioritäten werden auf die Prioritätsstufen des Echtzeitbetriebssystems abgebildet (siehe Abschnitt Task-Prioritäten, Seite 125).

Wenn die Watchdog-Zeit einer zyklischen Task höher ist als die eingestellte Intervallzeit und die Ausführung der Task nicht beendet ist, bevor die eingestellte Intervallzeit erreicht wird, werden ein oder mehrere Ausführungs-Zyklen übersprungen.

9.2.2 Ereignis-Task

Ereignis-Tasks oder auch Event-Tasks werden vom Betriebssystem gestartet, wenn bestimmte Ereignisse auftreten. Gegenwärtig sind folgende Ereignisse definiert:

Interne Bezeichnung	Ereignis-Nr.	Bemerkungen
Interrupts		
PLC_EVENT_XFIO_I0	0x00	XFIO Interrupt (Input 0, XCx3/4/5/8)
PLC_EVENT_XUIO_0	0x00	U-Bus Interrupt 0 (XCx4/7/8/11, UBE32 0,1l Eingang 0)
PLC_EVENT_XFIO_I1	0x01	XFIO Interrupt (Input 1, XCx3/4/5/8)
PLC_EVENT_XUIO_1	0x01	U-Bus Interrupt 1 (XCx4/7/8/11, UBE32 0,1l Eingang 1)
PLC_EVENT_XUIO_2	0x02	U-Bus Interrupt 2 (XCx4/7/8/11, UBE32 0,1l Eingang 2)
PLC_EVENT_XUIO_3	0x03	U-Bus Interrupt 3 (XCx4/7/8/11, UBE32 0,1l Eingang 3)
Synchronisation		
PLC_EVENT_POS	0x04	Lageregler-Task (nur XCN)
PLC_EVENT_CAN	0x05	- CANopen-Task, - gilt auch für Profibus-Task (microLine, XCx micro)
PLC_EVENT_IPO	0x06	CNC IPO-Task (nur XCN)
PLC_EVENT_DECO	0x07	CNC DECO-Task (nur XCN)
PLC_EVENT_MCSIO	0x08	MCS / XCS20 IO-Treiber Synchronisation (microLine, XCx micro)
Reserviert	0x09	
PLC_EVENT_XFIO_I10	0x0A	Messinterrupt aktiv 0 (nur XCx3/4/5/8)
PLC_EVENT_XFIO_I11	0x0B	Messinterrupt aktiv 1 (nur XCx3/4/5/8)
Reserviert	0x0C	
Reserviert	0x0D	
Reserviert	0x0E	
PLC_EVENT_AC_FAIL	0x0F	AC Fail (XCx 11, ProNumeric)

Tabelle 64: Multi-Task-System, Ereignis-Tasks

Die Ereignisnummer wird in der Taskeinstellung von MULTIPROG verwendet, um das Ereignis zu spezifizieren, das die Ereignis-Task startet.

Die vorgegebene Priorität wird, außer bei gesetzter Bypass-Option, vom System berücksichtigt. (Bypass hebt den normalen Taskwechsel auf, sodass die zugewiesenen Programme sofort ausgeführt werden, wenn das Ereignis eintritt.)

Es werden bis zu 16 Ereignisse in eine Warteschlange gesetzt. Diese Ereignisse gehen daher nicht verloren und werden später ausgeführt. Dies gilt auch im Falle eines Auftretens neuer Ereignisse vor der Ausführung der zugewiesenen Ereignis-Task.

9.2.3 System-Tasks

System-Tasks bzw. Systemprogramme (SPGs) werden automatisch vom Betriebssystem gestartet, wenn im Zusammenhang mit dem Betriebssystem ein Ereignis auftritt. Verschiedene SPGs sind verfügbar, wie in der folgenden Tabelle dargestellt:

Nr.	Name	Ereignis	Aktionen
SPG 0	**WARM_START**	wird bei einem Warmstart ausgeführt	• remanente Daten werden nicht initialisiert • nicht-gepufferte Daten werden initialisiert • die Open-Funktion des I/O-Treibers wird ausgeführt • Anwender-Tasks werden aktiviert • SPS wechselt in den Zustand 'Betrieb'
SPG 1	**COLD_START**	wird bei einem Kaltstart ausgeführt	• alle Daten werden initialisiert • die Open-Funktion des I/O-Treibers wird ausgeführt • Anwender-Tasks werden aktiviert • SPS wechselt in den Zustand 'Betrieb'
SPG 2	**TO_STOPP**	wird ausgeführt, wenn die Programmausführung gestoppt wird	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOPP'
SPG 10	**WATCHDOG**	wird ausgeführt, wenn die Ausführung einer Task nicht innerhalb ihrer Watchdogzeit beendet ist	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOPP'
SPG 11	**ZERODIV**	wird ausgeführt, wenn während der Programmausführung eine Division durch Null aufgetreten ist	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOPP'
SPG 12	**STACKOVER**	wird ausgeführt, wenn ein Stacküberlauf aufgetreten ist. Wird nur ausgeführt, wenn das Kontrollkästchen 'Stack-Prüfung' im Dialog 'Ressource ... einrichten' in MULTIPROG aktiviert wurde.	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOPP'
SPG 13	**BADCAL**	wird ausgeführt, wenn eine herstellerspezifische POE aufgerufen wird, die nicht existiert	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOPP'
SPG 14	**IOERROR**	wird ausgeführt, wenn ein Fehler im I/O-Treiber auftritt, während der Prozess abläuft	• SPS setzt Abarbeitung fort
SPG 16	**MATHERR**	wird ausgeführt, wenn ein Gleitkommafehler in einer arithmetischen Funktion auftritt	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOPP'

Nr.	Name	Ereignis	Aktionen
SPG 17	CPU_OVERLOAD	wird ausgeführt, wenn eine CPU-Überlastung auftritt	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOPP'
SPG 18	INITIODRV_ERR	wird ausgeführt, wenn beim Initialisieren des I/O-Treibers während eines Kalt- oder Warmstarts ein Fehler auftritt	• SPS wird nicht gestartet
SPG 19	BOUNDS_ERR	wird ausgeführt, wenn die Grenzen eines Felds oder einer Struktur überschritten wurden. Wird nur ausgeführt, wenn das Kontrollkästchen 'Index-Prüfung' oder das Kontrollkästchen 'Feldbegrenzungs-Prüfung' im Dialog 'Ressource ... einrichten' in MULTIPROG aktiviert wurde.	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOPP'
SPG 20	BUS_ERR	wird ausgeführt, wenn Variablen mit einem Datentyp ≥ 2 Bytes und ungeraden Adressen verwendet wurden oder wenn in MULTIPROG ein interner Fehler aufgetreten ist. Nur bei Motorola-Plattformen.	• Anwender-Tasks werden deaktiviert • alle Ausgänge werden aktualisiert • die Close-Funktion des I/O-Treibers wird ausgeführt • SPS wechselt in den Zustand 'STOPP'
SPG 21	STRING_ERR	wird ausgeführt, wenn ein Fehler bei einer Zeichenfolge-Operation auftritt, z.B. wenn eine Zeichenfolge durch eine andere ersetzt werden sollte, aber nicht gefunden wurde.	• Das Verhalten einer Zeichenfolge-Ausnahme hat sich geändert! In der Standardeinstellung wird nach dem Auftreten einer Zeichenfolge-Ausnahme das SPG 21 aufgerufen. Zusätzlich wird ein Eintrag in den Fehlerkatalog vorgenommen, der die Modul- und Zeilennummer enthält. Die SPS bleibt im 'RUN'-Status.

Tabelle 65: Multi-Task-System, System-Tasks

Hinweis

System-Tasks werden nicht vom Watchdog überwacht.

9.2.4 Default-Task

Die Default-Task läuft mit der niedrigstmöglichen Anwender-Task-Priorität als sog. Hintergrund-Task und ist nicht zeitüberwacht. Sie wird dann aktiviert, wenn alle höherprioren Anwender-Tasks abgearbeitet wurden. Die Default-Task ist dabei so konfiguriert, dass sie nur einen Teil der zur Verfügung stehenden Restzeit beansprucht. In jeder Ressource ist nur eine Default-Task erlaubt. Es wird empfohlen, ausschließlich zyklische Tasks zu verwenden.

Hinweis

Alle Treiber der I/O-Konfiguration, die nicht explizit einer Anwender-Task zugeordnet wurden, führen zum automatischen Anlegen der Default-Task und werden im Kontext der Default-Task ausgeführt.

9.2.5 Anwender-Task-Information

Für jede Anwender-Task werden Informationen auf System-Variablen abgebildet. Die unten abgebildeten Typdefinitionen der Systemvariablen sind in der Bibliothek "SchleicherLib" im Abschnitt *PLC_Types* zu finden.

Die Variablen werden mit den Typen *TaskInfoType0* und *TaskInfoType1* deklariert (→ Bild 83).

Bild 83:
Multi-Task-System, Variablendeklaration

Global_Variables:Configuration.Resource - Configuration.Resource.Global_Variables

Name	Typ	Verwendung	Beschreibung	Adresse	Anfangsw...	Reman...
TaskInfo0	TaskInfoType0	VAR_GLOBAL		%MD 1.1000		
TaskInfo1	TaskInfoType1	VAR_GLOBAL		%MD 1.1004		
TaskInfo2	TaskInfoType1	VAR_GLOBAL		%MD 1.1068		
TaskInfo3	TaskInfoType1	VAR_GLOBAL		%MD 1.1132		
TaskInfo4	TaskInfoType1	VAR_GLOBAL		%MD 1.1196		
TaskInfo5	TaskInfoType1	VAR_GLOBAL		%MD 1.1260		

Die folgende Anwendertask-Information wird mit einem Offset von 64 ab 1004 deklariert (1004 + 64 = 1068 usw.).

Die Reihenfolge der Tasks wird durch den Rang der Task im Projektbaum *Physikalische Hardware/Configuration/Ressource/Tasks* festgelegt.

9.2.6 Task-Prioritäten

Die Tabelle gibt eine Übersicht über die empfohlenen Task-Prioritäten bzw. deren Einordnung hinsichtlich wichtiger reservierter Firmware-Tasks (*tfwLAGE, tfwCANhigh, tfwIPO*).

Warnung!

Eine falsche oder zumindest ungeeignete Wahl der Anwender-Task-Einstellungen hinsichtlich Typ, Priorität oder Interrupt-Mode usw. – insbesondere in Kombination mit langen Programmlaufzeiten – kann zu Steuerungsfehlfunktionen führen, da essentielle Firmware-Tasks (*tfwLAGE, tfwCANhigh, tfwIPO*) verdrängt werden.

Die Taskzuordnung und Taskzeiteinstellung überprüfen bzw. anpassen.

MULTIPROG-Priorität	RTOS*-Priorität (default)	RTOS* Task-Name	Verwendung
0	30	beliebig	z.B. Anwender-Task (Ereignis 0)
1	31	beliebig	z.B. Anwender-Task (Ereignis 1)
2	32	beliebig	z.B. Anwender-Task (Ereignis 4)
3	33	*tfwLAGE*	reserviert für Lageregel-Task (nur XCN)
4	34	beliebig	z.B. Anwender-Task (Ereignis 4, 5)
5	35	*tfwCANhigh*	reserviert für CAN- Stack Task (Option CAN_HIGH_PRIO = 1)
6	36	beliebig	z.B. Anwender-Task (Ereignis 5, 6)
7	37	*tfwIPO*	reserviert für IPO-Task (nur XCN)
8	38	beliebig	z.B. Anwender-Task (Ereignis 5)
9	39	*tfwCANhigh*	reserviert für CAN-Stack-Task (Option CAN_HIGH_PRIO = 0)
10	40	beliebig	z.B. Anwender-Task (Ereignis 5)
11..15	41..45	beliebig	z.B. zyklische Anwender-Tasks
16..31	46	beliebig	z.B. sonstige, zyklische Anwender-Tasks
Default	127	*default*	Hintergrund-Task
			*__R__eal __T__ime __O__perating __S__ystem

Tabelle 66: Multi-Task-System, Taskprioritäten

Hinweis

Das System unterstützt insgesamt 18 Anwender-Tasks (Prioritätsstufen 0..16 und die Default-Task). Tasks mit Prioritätswerten ≥ 16 werden mit Priorität 16 ausgeführt.

9.2.7 Task und Watchdog

Es gibt zu jeder anwenderdefinierten Task einen eigenen einstellbaren Watchdog. Der Watchdog überprüft, ob die Taskausführung am Ende des Watchdog-Zeitintervalls beendet ist. Wenn die Taskausführung nach dieser Zeit nicht beendet wird, wird die System-Task *SPG 10 'WATCHDOG'* ausgeführt und die SPS geht in den *'STOPP'*-Zustand über, wenn keine weiteren Aktionen programmiert wurden. Zusätzlich wird ein Eintrag in den Fehlerkatalog vorgenommen. Das Watchdog-Zeitintervall beginnt, wenn die Task bereit für die Ausführung ist. Das Watchdog-Zeitintervall wird im Dialog *'Task ... einrichten'* in MULTIPROG festgelegt.

Hinweis

Wenn die Ausführungsdauer der Task sowie die Watchdog-Zeit annähernd denselben Wert haben und eine hohe CPU-Auslastung vorliegt, ist es möglich, dass während der Umsetzung einiger Online-Bedienschritte die Watchdog-Zeit überschritten wird.

Ein Grund für dieses Verhalten kann sein, dass während des Debuggens im Online-Modus der Adressstatus mit Durchlaufkontrolle ausgewählt wurde.

Beispiel

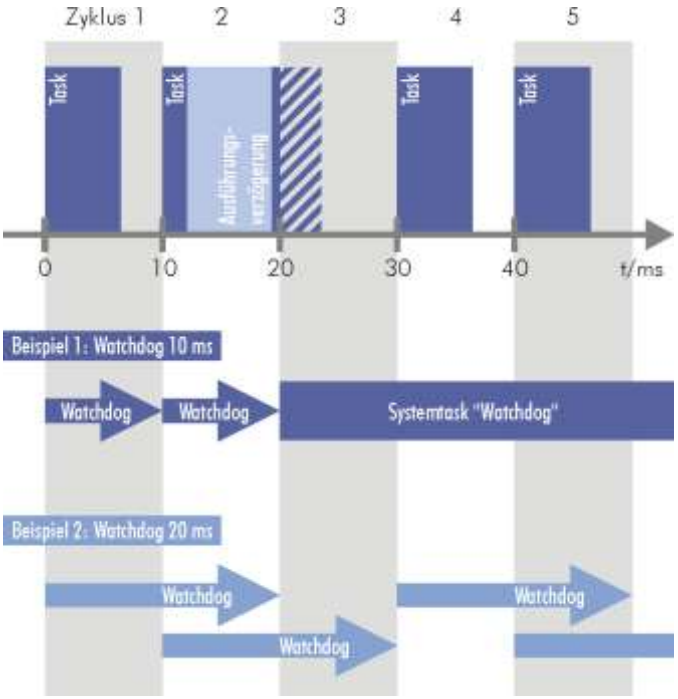


Bild 84: Multi-Task-System, Beispiel für Tasks und Watchdogs

Im **Beispiel 1** ist die Watchdog-Zeit der angezeigten Task auf 10 ms eingestellt. In der Abbildung überschreitet sie ihre Watchdog-Zeit im zweiten Zyklus nach 20 ms. Die Ausführung der Task wird abgebrochen und die Systemtask "Watchdog" aufgerufen.

Im **Beispiel 2** ist die Watchdog-Zeit auf 20 ms eingestellt. Er spricht deshalb auf die Zeitüberschreitung der Task im zweiten Zyklus nicht an. Die Task setzt lediglich für den nächsten Zyklus aus und wird erst wieder nach 30 ms im vierten Zyklus ausgeführt.

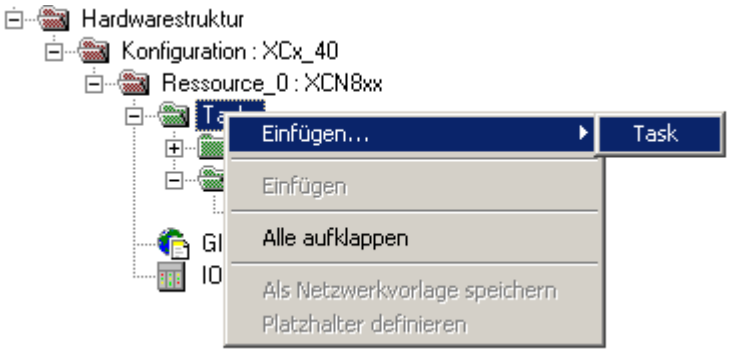
9.2.8 Task einfügen und Programm zuweisen

Task einfügen

Um eine Task einzufügen, müssen in MULTIPROG folgende Schritte ausgeführt werden:

- Im Projektbaum, unter der Ressource für die jeweilige Steuerung, mit der rechten Maustaste auf den Ordner *Tasks* klicken, um das Kontextmenü zu öffnen (→ Bild 85).

Bild 85: Multi-Task-System, Einfügen einer Task in MULTIPROG



- Den Menüpunkt *Einfügen/Task wählen*, es erscheint der Dialog *Einfügen*.
- Den Namen für die Task eingeben.
- Im Listenfeld *Task-Typ* den gewünschten Task-Typ einstellen. Es kann zwischen einer Default-Task, zyklischen Task, der Ereignis- oder System-Task gewählt werden. Hinweis: Wenn der Tasktyp 'DEFAULT' nicht in dem Listenfeld aufgeführt ist, besitzt die Ressource bereits eine Default-Task.
- Den Dialog mit *OK* bestätigen.

Es erscheint der Dialog *Task-Einstellungen für ...* Abhängig von der zuvor ausgewählten Task enthält der Dialog verschiedene Text- und Listenfelder.

Für die jeweilige Task müssen folgende Parameter eingegeben werden:

Tabelle 67: Multi-Task-System, Taskparameter

Task	Parameter
zyklische Task	Zeitintervall
Ereignis-Task	Ereignisnummer (Nummer des Interrupt)
System-Task	Nummer eines Systemprogrammes

Bei der Vergabe der Priorität müssen unbedingt die Ausführungen im Abschnitt Task-Prioritäten, Seite 125 beachtet werden.

Programme müssen Tasks zugewiesen werden, um sie auszuführen. Zuweisen eines Programmes zu einer Task bedeutet, dass eine Instanz des Programmes ausgeführt wird, wenn die Task aktiviert wird. Von einem Programm können verschiedene Instanzen verschiedenen Tasks zugeordnet werden.

Einer Task können mehrere Programme zugewiesen werden. In diesem Fall wird das erste Programm im Taskverzeichnis als erstes ausgeführt. Danach wird das Programm darunter ausgeführt usw.

Programme zuweisen

Um Programme zuzuweisen, müssen in MULTIPROG folgende Bedienschritte durchgeführt werden:

- Im Projektbaum mit der rechten Maustaste auf das Symbol der Task klicken, in die das Programm eingefügt werden soll (→ Bild 86).
- Im Kontextmenü *Einfügen* / *Programminstanz* wählen.
- Einen Instanznamen für das Programm in das Feld *Programminstanz* eingeben.
- Im Listenfeld *Programmtyp* das gewünschte Programm einstellen.
- Den Dialog mit OK bestätigen.

Das Programmsymbol wird in den Projektbaum eingefügt.

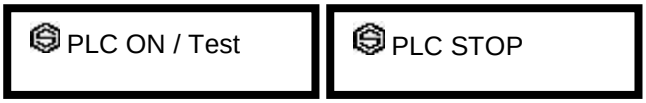
Bild 86: Multi-Task-System, Zuweisen von Programmen in MULTIPROG

10 Displayfunktionen der XCx800

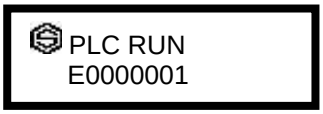
Allgemeine Parameter:
graphisch, 128 x 32 Pixel
ausreichend für 2 Zeilen zu 16 Zeichen

10.1 Displayanzeige des Betriebszustands

Es werden die Betriebszustände der Steuerung inkl. etwaiger Fehler angezeigt, z.B.



falls sich Fehler im „Active Errorbuffer“ befinden (keine Warnungen!) werden diese in der unteren Zeile zusätzlich angezeigt



mit Hilfe der RESET Taste können weitere Funktionen angewählt werden (siehe Tabelle Display Funktionen).

z.B. kann alternativ der Fehlertext statt der Fehlernummer ausgegeben werden

Achtung:

Der Taster RESET löst nur in Schalterstellung 0 einen HW Reset aus!

10.2 Display Funktionen in Abhängigkeit vom Betriebsartenschalter

Stellung	Anzeige wenn	Funktion
0	--	aktuelle IP Adresse und MAC ID
1	--	Anzeige der Fehlernummer des letzten, „aktiven“ Fehlers aus dem „Active Error Buffer“
	RESET gedrückt	aktuelle XCx-, ProConOS – und Koppelspeicher- Version, Bootprojekt
2	--	Anzeige der Fehlernummer des letzten, „aktiven“ Fehlers aus dem „Active Error Buffer“
	RESET gedrückt	Anzeige des Fehlertextes des letzten, „aktiven“ Fehlers aus dem „Active Error Buffer“
3	--	Anzeige der Fehlernummer des letzten, „aktiven“ Fehlers aus dem „Active Error Buffer“
	RESET gedrückt	(wechselnde) Anzeige aller aktiven Fehler aus dem „Active Error Buffer“
4		Anzeige des Fehlertextes des letzten, „aktiven“ Fehlers aus dem „Active Error Buffer“
	RESET gedrückt	Anzeige der Fehlernummer des letzten, „aktiven“ Fehlers aus dem „Active Error Buffer“
5		keine Fehleranzeige
	RESET gedrückt	(wechselnde) Anzeige aller aktiven Fehler aus dem „Active Error Buffer“
6		keine Fehleranzeige
	RESET gedrückt	(wechselnde) Anzeige aller aktiven Fehler aus dem „Active Error Buffer“
7		keine Fehleranzeige (für Nutzung durch einen speziellen FB SPS, i.V.)
	RESET gedrückt	keine Fehleranzeige (für Nutzung durch einen speziellen FB SPS, i.V.)
8		Anzeige der Fehlernummer des letzten, „aktiven“ Fehlers aus dem „Active Error Buffer“
	RESET gedrückt	Anzeige zusätzlicher Informationen: Seriennummer und Typcode
9		Anzeige der Fehlernummer des letzten, „aktiven“ Fehlers aus dem „Active Error Buffer“
	RESET gedrückt	Anzeige zusätzlicher Informationen: - CANopen Node ID und Baudrate, - SERCOS III IP-Adresse / MAC-ID, - aktueller „Rootpfad“ des Dateisystems, etc.

Tabelle 68: Display Funktionen

11 Die RS232-Schnittstelle am Service-PC einrichten

Die virtuelle Schnittstelle ist dem Echtzeitbetriebssystem zugeordnet und dient zum Anschluss von Bediengeräten und zur Systemdiagnose. Auf ihr erfolgt z.B. die Ausgabe des Bootprotokolls beim Hochfahren der XCx.

- Die Kabelverbindung zwischen dem Service-PC COM1 oder COM2 und dem XCx-Anschluss X7 herstellen (Pinbelegung → Seite 18).
- Auf dem PC unter *Start / Programme / Zubehör / Kommunikation* das Programm *HyperTerminal* starten, einen Namen eingeben (z.B. XCx) und ein Symbol aussuchen.
- Im Dialogfenster *Eigenschaften von <Name>* unter *Verbindung herstellen über* die Direktverbindung über COM1 auswählen und dann auf den Button *Konfigurieren* klicken (→Bild 87).

Bild 87: Dialogfenster "Eigenschaften von XCX", Auswahl der Verbindung

- In *Eigenschaften von COM1* diese Parameter einstellen (→ Bild 88).
  - Bits pro Sekunde: 115200
  - Datenbits: 8
  - Parität: Keine
  - Stoppbits: 1
  - Flusssteuerung: Kein

Bild 88: Dialogfenster "Eigenschaften von COM1", Anschluss-einstellungen

- Die XCx einschalten oder RESET auslösen.

Während des Steuerungsanlaufes wird im HyperTerminal-Dialogfenster das Bootprotokoll ausgegeben.

11.1 Auslesen der IP-Adresse der XCx über die RS232-Schnittstelle

XCx800:
Die eingestellte IP-Adresse wird im Display angezeigt, Mode-Schalter in Stellung 0.

XCx400:

Im Dialogfenster von *Hyperterminal* den Befehl *version* eingeben.

Es wird die Versionsinformation ausgegeben.

```
-> version
VxWorks (for Freescale iMx31 - ARM1136JFS) version
6.7.
Kernel: WIND version 2.12.
Made on Nov 17 2010, 18:02:24.
Boot line:
ata=0,0(0,0):/ata0/OS/XCx800/VxWorks.xcn
e=192.168.1.2:ffffff00 u=schleicher pw=berlin f=0x8
tn=XCx s=/ata0/OS/run.vxs o=sme
value = 139 = 0x8b
```

Die IP-Adresse (192.168.1.2), Subnet Mask (255.255.255.00), Username (u=schleicher) und Passwort (pw=berlin) stehen in der Zeile:

```
e=192.168.1.2:ffffff00 u=schleicher pw=berlin f=0x8 tn=XCx
```

12 Die CNC

Die XCN 800 ist eine CNC mit bis zu 32 Achsen/Spindeln und integrierter, leistungsfähiger SPS.

Funktionsübersicht

- Bis zu 16 Teilsystemen mit insgesamt 32 Achsen/Spindeln
- Technologien für Bohren, Fräsen, Schleifen, Handling
- Endlosdrehende Rundachsen
- Spindelpaket mit umfangreicher Funktionalität, wie z.B. Gewindeschneidfunktionen, variable Pulsbewertung, orientierter Spindelhalt
- Synchronspindel
- Programmierbare Beschleunigung
- Elektronisches Getriebe
- 2D+n-Schraubenlinien-Interpolation
- Vorschub und Eilgang :
  0,001 mm/min bis 999 m/min
- Werkzeugradiuskorrektur mit An- und Abfahrstrategien
- Werkzeuglängenkorrektur
- Interpolatorische Spindelsteigungs- und Messsystemfehlerkompensation
- Losekompensation
- Arbeitsfeldbegrenzung
- Softwareendschalter

Die Inbetriebnahme der CNC-Funktionen und die CNC-Programmierung der XCN werden ausführlich in einer eigenen Betriebsanleitung beschrieben (→ Seite 10).

13 Anhang

13.1 Technische Daten aller Module

Klimatische Bedingungen	
Betriebsumgebungstemperatur	0 ... +55°C (Kl. KV nach DIN 40040), senkrechter Einbau, freie Luftzirkulation
Lagertemperatur	-25 ... +70°C (Kl. HS nach DIN 40040)
Relative Luftfeuchte	10 ... 95% (Kl. F nach DIN 40040), keine Betauung
Luftdruck im Betrieb	860 ... 1060 hPa

Mechanische Festigkeit	
Schwingen	nach DIN EN 60068-2-6 10 ... 57 Hz konstante Amplitude 0,075mm 57 ... 150 Hz konstante Beschleunigung 1 g
Schocken	nach DIN EN 60068-2-27, Sinus-Halbwelle 15g / 11ms
Freier Fall	nach DIN EN 60068-2-32, Fallhöhe 1m (mit Originalverpackung)

Elektrische Sicherheit	
Schutzklasse	Klasse I nach EN 60536 (Basisisolierung und Schutzleiteranschluss)
Schutzart	IP 00 nach EN 60529
Luft-/Kriechstrecken	nach DIN EN 61131-2, zwischen Stromkreisen und Körper sowie zwischen galvanisch getrennten Stromkreisen, entsprechend Überspannungskategorie II, Verschmutzungsgrad 2
Prüfspannung	AC 350 V/50Hz für Geräte-Nennspannung DC 24V AC 1350 V/50Hz für Geräte-Nennspannung AC 230V

Elektromagnetische Verträglichkeit	
Elektrostatische Entladung	nach EN 61000-4-2, 8 KV Luftentladung, 4 KV Kontaktentladung
Elektromagnetische Felder	nach EN 61000-4-3, Feldstärke 10 V/m, 80 ... 1000 MHz
Schnelle Transienten (Burst)	nach EN 61000-4-4, 2 KV auf AC/DC-Versorgungsleitungen, 1 KV auf E/A-Signalleitungen
Störaussendung	nach EN 55011, Grenzwertklasse A, Gruppe 1

13.2 Zubehör und Software

Bezeichnung	Beschreibung	Artikelnummer
MULTIPROG 4.x	SPS Programmiersystem nach IEC61131-3	R4.320.0640.0
Service Pack	Steuerungssoftware für alle Schleicher-Steuerungen, Add-Ons, Schleicher-Dialog, Dokumentationen und Service-Informationen	R4.320.0590.0
ProCANopen	Netzwerk-Konfigurationssoftware	R4.320.0500.0
CANcardY	1-fach CANopen-Interface, PCMCIA-Steckkarte	R4.321.0020.0

Tabelle 69: Zubehör und Ersatzteile

13.3 Warenzeichenvermerke

- WINDOWS ist ein eingetragenes Warenzeichen der Microsoft Corporation.
- CANopen ist ein eingetragenes Warenzeichen von CAN in Automation e.V,
- ProCANopen ist ein eingetragenes Warenzeichen von Vector Informatik GmbH.
- VxWorks ist ein eingetragenes Warenzeichen der Wind River Systems Inc.
- PROFIBUS ist ein eingetragenes Warenzeichen der PROFIBUS Nutzerorganisation.
- MULTIPROG ist ein eingetragenes Warenzeichen der KW-Software GmbH

Alle anderen Warenzeichen oder Produktnamen sind eingetragene Warenzeichen der jeweiligen Firmen.

13.4 Abbildungsverzeichnis

Bild 1: Systemübersicht XCx 400 / 800, Erweiterungsmodule und Peripherie 11
Bild 2: Aufbau des Gesamtsystems 12
Bild 3: Montage der Module auf dem Baugruppenträger 13
Bild 4: Steuereinheit XCx 800 14
Bild 5: Steuereinheit XCx 400 15
Bild 6: Display 23
Bild 7: Installation von MULTIPROG 31
Bild 8: Installation des OPC-Servers 32
Bild 9: Installation der Add-Ons 33
Bild 10: AddOn-Installation, Eingabe der ProCANopen-Version 33
Bild 11: Installation des Schleicher-Dialogs 34
Bild 12: Kommunikationsschema 35
Bild 13: Ethernet Verbindung prüfen 36
Bild 14: IP-Adresse ändern im Schleicher Dialog 37
Bild 15: Aufruf der Ressource-Einstellungen in MULTIPROG 38
Bild 16: Eingeben der IP-Adresse in MULTIPROG 39
Bild 17: Aufruf der Verbindung zur XCx 400 / XCx 800 in MULTIPROG 39
Bild 18: Ressource-Fenster bei Online-Verbindung zur XCx 800 40
Bild 19: Neues Projekt in MULTIPROG öffnen 41
Bild 20: Steuerungstyp für neues Projekt auswählen 41
Bild 21: Beispiel für Projektbaum in MULTIPROG 42
Bild 22: Speicherdialog aufrufen 42
Bild 23: Dialogfenster "Projekt speichern" 42
Bild 24: Projekt kompilieren 43
Bild 25: Projekt auf die Steuerung übertragen 43
Bild 26: Online-Anzeige der Variablen 44
Bild 27: Aufruf des Dialogfensters "SPS/MC/CNC Koppelspeicher" 45
Bild 28: Dialogfenster "SPS/CNC/MC Koppelspeicher" 46
Bild 29: Auslesen der Betriebsoftware-Version 47
Bild 30: Darstellung der Koppelspeicher-Fehlermeldung im Schleicher-Dialog 48
Bild 31: Menüeintrag "Extras / XIO Konfiguration" 49
Bild 32: Dialogfenster "XRIO-Konfiguration / XUIO-Konfiguration" 49
Bild 33: Klemme X10, Funktionsauswahl 52
Bild 34: Klemme X11, Anschlüsse 1-4, Funktionsauswahl 52
Bild 35: Klemme X11, Anschlüsse 5-8, Funktionsauswahl 52
Bild 36: XRIO-Konfiguration, Standardfunktionalität 56
Bild 37: XRIO-Konfiguration, erweiterte Funktionalität 56
Bild 38: Dialogfenster "XUIO-Konfiguration" 59
Bild 39: Knoten "IO_Configuration" 60
Bild 40: Dialogfenster "I/O-Konfiguration" 60
Bild 41: Knoten "Global_Variables" 61
Bild 42: Einfügen der globalen Variablen in das SPS-Projekt 61
Bild 43: Dialogfenster "XUIO-Konfiguration", Modul hinzufügen 62
Bild 44: Dialogfenster "XUIO-Konfiguration", Eigenschaften 62
Bild 45: Dialogfenster "Modul-Eigenschaften", Einstellen der Optionen 63
Bild 46: Knoten für Interrupt-freigabe und -verarbeitung 74
Bild 47: Globale Variablen für Interruptzählung und Kommunikation 74
Bild 48: Zyklische Task "XUIO" mit POE "X8_Interrupt_IO" 76
Bild 49: Einstellungen für zyklische Task 76
Bild 50: Anlegen der Event-Tasks für die Interrupts 77
Bild 51: Einstellungen für Event-Tasks 77
Bild 52: Einstellungen für Event-Tasks 77
Bild 53: Anschlussprinzip für CANopen-Netzwerk 79
Bild 54: DIP-Schalter am Kompaktmodul RIO 8 I/O CANopen 80
Bild 55: CANopen-Konfiguration, Dialogfenster "I/O-Konfiguration" 80
Bild 56: CANopen-Konfiguration, Dialogfenster "Eigenschaften" 81
Bild 57: CANopen-Konfiguration, Treiberparameter einstellen 81
Bild 58: CANopen-Konfiguration, Arbeitsblatt "Network_Variables" 82
Bild 59: CANopen-Konfiguration, Arbeitsblatt "I/O_Variables" 82
Bild 60: Aufruf der "Schleicher CANopen Konfiguration" in MULTIPROG 83
Bild 61: Schleicher CANopen Konfiguration 83
Bild 62: Aufruf von "ProCANopen Konfigurations-werkzeug" in MULTIPROG 86
Bild 63: Auswahl der CAN-Karte und Starten von ProCANopen 86
Bild 64: ProCANopen mit aktuellem CANopen-Projekt 86
Bild 65: Einlesen des Netzes mit ProCANopen 87
Bild 66: Einstellen der Scan-Optionen 87

Bild 67: Graphische Verknüpfung der Netzwerk-Knoten 88
Bild 68: Kommunikations beziehung zwischen Steuerung und I/O-Modul 88
Bild 69: Auswahl des Konfigurationsmanagers 89
Bild 70: Speichern der CAN-Konfiguration in den Konfigurationsmanager 89
Bild 71: Test der Netzwerkverbindung mit POE "CANdemo" 90
Bild 72: POE CANdemo mit dazugehörigem Variablen-Arbeitsblatt 90
Bild 73: Deklaration von PDD-Variablen in MULTIPROG 92
Bild 74: Deklaration der OPC-Variablen 95
Bild 75: Schleicher-Dialog, Startfenster 101
Bild 76: Schleicher-Dialog, Status- und Meldebereich 101
Bild 77: Koppelspeicher als Verbindung von SPS und CNC 106
Bild 78: MULTIPROG, Wechseln der Betriebszustände 108
Bild 79: MULTIPROG, Kontextmenü "Bibliotheken" 111
Bild 80: MULTIPROG, Dialogfenster "Bibliothek einbinden" 112
Bild 81: MULTIPROG, Einbinden der Bibliothek "Serial" 112
Bild 82: Multi-Task-System, Prioritätsstufen 119
Bild 83: Multi-Task-System, Variablendeklaration 124
Bild 84: Multi-Task-System, Beispiel für Tasks und Watchdogs 126
Bild 85: Multi-Task-System, Einfügen einer Task in MULTIPROG 127
Bild 86: Multi-Task-System, Zuweisen von Programmen in MULTIPROG 128
Bild 87: Dialogfenster "Eigenschaften von XCX", Auswahl der Verbindung 131
Bild 88: Dialogfenster "Eigenschaften von COM1", Anschluss-einstellungen 132

13.5 Tabellenverzeichnis

Tabelle 1: Zusätzliche Betriebsanleitungen 10
Tabelle 2: Pinbelegung der sercos III-Schnittstellen X1, X2 16
Tabelle 3: Pinbelegung der Ethernet-Schnittstellen X3 (RJ 45) 16
Tabelle 4: Pinbelegung der USB-Schnittstellen X4 16
Tabelle 5: Pinbelegung der USB-Schnittstellen X5 17
Tabelle 6: Pinbelegung der CAN-Schnittstellen X6 17
Tabelle 7: Pinbelegung der RS 232 / 422 / 485-Schnittstellen X7 18
Tabelle 8: Pinbelegung der Betriebsspannung / Betriebsbereitrelais -Schnittstellen X7 18
Tabelle 9: Pinbelegung der 24V I/ 19
Tabelle 10: Pinbelegung der 5V Eingangs -Schnittstellen X10 19
Tabelle 11: Pinbelegung der 5V Ausgangs -Schnittstellen X11 20
Tabelle 12: Anschluss der Schnittstelle X11 an den Buskoppler RIO EC X2 20
Tabelle 13: LED-Anzeigen auf der Steuerung 21
Tabelle 14: LED-Anzeigen an der Ethernet- Buchse (1x RJ45) 22
Tabelle 15: LED-Anzeigen an den sercos-III-Buchsen (2x RJ45) 22
Tabelle 16: Betriebsartenschalter 23
Tabelle 17: Reset-Taster 23
Tabelle 18: Liste der verfügbaren 29
Tabelle 19: Inhalt der CD-ROM 30
Tabelle 20: Systemvoraus setzungen 30
Tabelle 21: Bezeichnungen der Netzwerkeinstellungen 38
Tabelle 22: XFIO-Auswahl 51
Tabelle 23: XRIO- Auswahl 53
Tabelle 24: XRIO- Eigenschaften 54
Tabelle 25 xrio1_bc<n> 56
Tabelle 26: XUIO- Auswahl 58
Tabelle 27: Modultypen 64
Tabelle 28: Modulklassen 64
Tabelle 29: XFIO-Parameter 65
Tabelle 30: Eingangsbits xfio_IW0 66
Tabelle 31: Ausgangsbits xfio_QW0 66
Tabelle 32: Inkrementalwertgeber -Parameter 67
Tabelle 33: xfio_MW164_INC_ control 67
Tabelle 34: xfio_MW178_INC_status 67
Tabelle 35 SPS-Variablenliste Inkrementalwertgeber 68
Tabelle 36: Absolutwertgeber-Parameter 69
Tabelle 37: xfio_MW228_ABS _control 69
Tabelle 38: xfio_MW242_ABS_status 70
Tabelle 39 SPS-Variablenliste Absolutwertgeber 70
Tabelle 40: PWM-Variablen 72
Tabelle 41 SPS-Variablenliste PWM 73
Tabelle 42 SPS-Variablenliste Interrupt 74
Tabelle 43: Pinbelegung der verwendeten Stecker 79
Tabelle 44: Einstellen von Knotennummer und Datenübertragungsrate 80
Tabelle 45: Schleicher-Dialog, Aufteilung der Bedienoberfläche 101
Tabelle 46: Schleicher-Dialog, Status- und Meldebereich 101
Tabelle 47: SPS-Betriebszustände 107
Tabelle 48: MULTIPROG, Starten der Programmausführung 108
Tabelle 49: MULTIPROG, Stoppen der Programmausführung 109
Tabelle 50: MULTIPROG, Allgemeines Reset 109
Tabelle 51: Systemvariablen 110
Tabelle 52: Bibliotheken und Funktionsbausteine in MULTIPROG 111
Tabelle 53: Bibliothek CANopen_Vxxx 113
Tabelle 54: Bibliothek CFB_Vxxx 113
Tabelle 55: Bibliothek CNC_Vxxx 114
Tabelle 56: Bibliothek Date_Time 115
Tabelle 57: Bibliothek MC_Vxxx 115
Tabelle 58: Bibliothek MMI 116
Tabelle 59: Bibliothek PLC_Vxxx 116
Tabelle 60: Bibliothek PNS_Vxxx 117
Tabelle 61: Bibliothek Profibus_Vxxx 117
Tabelle 62: Bibliothek Serial 117
Tabelle 63: Bibliothek XCx7_Vxxx 118
Tabelle 64: Multi-Task-System, Ereignis-Tasks 121
Tabelle 65: Multi-Task-System, System-Tasks 123
Tabelle 66: Multi-Task-System, Taskprioritäten 125

Tabelle 67: Multi-Task-System, Taskparameter 127
Tabelle 68: Display Funktionen..... 130
Tabelle 69: Zubehör und Ersatzteile 135

13.6 Index

Abbildungsverzeichnis ... 136
Absolutwertgeber ... 52, 65, 69
Absolutwertgeber, SPS Beispiel ... 70
Active-Error-Buffer ... 59, 105
Add-Ons, Installation ... 33
Anschluss von RIO EC X2 (XRIO) an X11 ... 20
Anwenderbibliotheken ... *Siehe* Bibliotheken
Anwender-Task ... 120
Anwender-Task-Information ... 124
Applikationsbeispiele für XCx400 / XCx800 I/O Schnittstellen 65
Automatik ... 102
Baugruppenträger ... 28
Bedienung der XCx mit dem Schleicher Dialog ... 94
Bestellangaben XCx 400 / 800 ... 27
BETRIEB Betriebszustand ... 107
Betriebsartenschalter ... 23
Betriebsbereitrelais ... 18
Betriebsspannung ... 18
Betriebszustände der SPS ... 107
Bibliothek
  CANopen_Vxxx ... 113
  CFB_Vxxx ... 113
  CNC_Vxxx ... 114
  Date_Time ... 115
  MC_Vxxx ... 115
  MMI ... 116
  PLC_Vxxx ... 116
  PNS_Vxxx ... 117
  Profibus_VxxxSerial ... 117
  SchleicherLib_Vxxx ... 117
  Serial ... 117
  XCx7_Vxxx ... 118
CANopen
  Anschluss an die Steuereinheit ... 17
  Anschlussprinzip ... 79
  Configuration manager ... 84
  Deklaration des I/O-Treibers ... 80
  EDS-Datei ... 84
  Einstellungen am RIO-Modul ... 80
  Feldbuskarte ... 84
  Konfiguration mit ProCANopen ... 84
  NMT manager ... 84
  Schleicher CANopen Konfiguration ... 83
  Spezifikationen ... 78
CAN-Schnittstellen ... 17
CNC ... 133
Darstellungskonventionen ... 6
Default-Task ... 124
Deklaration
  OPC-Variable ... 95
Deklaration von Netzwerkvariablen ... 82
Die RS232-Schnittstelle am Service-PC einrichten ... 131
Display nur XCx 800 ... 23
Displayfunktionen
  Displayanzeige des Betriebszustands ... 129
  in Abhängigkeit vom Betriebsartenschalter ... 130
Displayfunktionen der XCx800 ... 129
Echtzeit-Uhr ... 115
EDS-Datei ... 84
EIN Betriebszustand ... 107
Einbau des Automatisierungssystems ... 12
Ereignis-Task ... 121
Error-Log-Book ... 59
Erweiterungsmodule ... 28
Ethernet
  Inbetriebnahme der Schnittstelle ... 36
  IP-Adresse Einstellen/Auslesen auf der XCx ... 132
  Schnittstellen ... 16
  TCP/IP-Einstellungen anpassen ... 37
  TCP/IP-Verbindung zur Programmierung ... 30
Fehlermeldungen ... 22
  Beispiel ... 59
Firmware-Task Prioritäten ... 125
Fremdsoftware starten ... 104
Frontansicht XCx 400 ... 15
Frontansicht XCx 800 ... 14
FTP ... 37
Funktionsauswahl
  Klemme X10 ... 52
  Klemme X11 ... 52
Funktionsbaustein
  CLEAR_ERROR ... 116
  CLOSE_PROFILE ... 116
  CO_NET_CAN_SYNC ... 113
  CO_NET_GET_KERNEL_STATUS ... 113
  CO_NET_GET_LOCAL_NODE_ID ... 113
  CO_NET_GET_STATE ... 113
  CO_NET_NMT ... 113
  CO_NET_PING ... 113
  CO_NET_RECV_EMY ... 113
  CO_NET_RECV_EMY_DEV ... 113
  CO_NET_RECV_ERR ... 113
  CO_NET_RECV_ERR_DEV ... 113
  CO_NET_RESTART_ALL ... 113
  CO_NET_RESTART_CAN ... 113
  CO_NET_SDO_READ ... 113
  CO_NET_SDO_WRITE ... 113
  CO_NET_SENDL2 ... 113
  CO_NET_SHUTDOWN ... 113
  CONNECT_V ... 113
  DP_NET_GET_MSG ... 117
  DP_NET_GET_STATE ... 117
  DP_NET_PUT_MSG ... 117
  FLUSH_PROFILE ... 116
  GET_DATE ... 115
  GET_MTS ... 116
  GET_PROFILE_INT ... 116
  GET_PROFILE_REAL ... 116
  GET_PROFILE_STRING ... 116
  GET_TIME ... 115
  IBSM ... 118
  MC_ANALOG ... 114
  MC_ANALOG_1_AXIS ... 114
  MC_CAN ... 114
  MC_CAN_1_AXIS ... 114
  MC_DP ... 114
  MC_DP_1_AXIS ... 114
  MC_GetCncAxis ... 115
  MC_Home ... 115
  MC_MoveAbsolute ... 115
  MC_MoveAdditive ... 115
  MC_MoveRelative ... 115
  MC_MoveVelocity ... 115
  MC_Power ... 115
  MC_ReadActualPosition ... 115
  MC_ReadAxisError ... 115
  MC_ReadBoolParameter ... 115
  MC_ReadParameter ... 115
  MC_ReadStatus ... 115
  MC_ReleaseCncAxis ... 115
  MC_Reset ... 115
  MC_Stopp ... 115

MC_WriteBoolParameter ... 115
MC_WriteParameter ... 115
NEW_PROFILE ... 116
OPEN_PROFILE ... 116
PNSReadIOData ... 117
PNSWriteIOData ... 117
PORT_CLOSE ... 117
PORT_OPEN ... 117
PORT_READ ... 117
PORT_STATE ... 117
PORT_WRITE ... 117
PPF_COP_COMM ... 116
PUT_ERROR ... 116
PUT_ERROR2 ... 116
READ_AXIS_PAGE ... 118
READ_FILE ... 116
READ_Q_PARAM_* ... 114
READ_RP ... 118
READ_SERC_PARAM ... 114
SAVE_Q_PARAM_* ... 114
SAVE_R_PARAM_* ... 114
SEND_MAIL ... 116
SET_DATE ... 115
SET_SERC_COMMAND ... 114
SET_SERC_PHASE ... 114
SET_TIME ... 115
UBA_ERR_CTRL ... 118
URCV_V ... 113
USEND_V ... 113
UZB_VR ... 118
WRITE_AXIS_PAGE ... 118
WRITE_FILE ... 116
WRITE_PROFILE_INT ... 116
WRITE_PROFILE_REAL ... 116
WRITE_PROFILE_STRING ... 116
WRITE_Q_PARAM_* ... 114
WRITE_RP ... 118
WRITE_SERC_PARAM ... 114
XFIO_CONFIG ... 116
XRIO_STATE ... 116
Funktionsbausteine und Bibliotheken ... 111
Grundinitialisierung ... 93
HALT Betriebszustand ... 107
Handbetrieb ... 102
Hardware-Konfiguration
Einlesen ... 59
I/O-Konfiguration einfügen ... 60
Optionen ... 61
High-Speed Ausgänge 5V DC ... 19
High-Speed Eingänge 5 V DC ... 19
HyperTerminal ... 131
I/O
X9, X10, X11 Bitbelegung ... 65
I/O-Ebene ... 49
I/O-Treiber CANopen ... 80
Inbetriebnahme ... 30, 104
Inbetriebnahme der Ethernet-Schnittstelle ... 36
Inkrementalwertgeber ... 52, 65, 67
Inkrementalwertgeber, SPS Beispiel ... 68
Installation ... 30
Add-Ons ... 33
MULTIPROG ... 31
OPC-Server ... 32
Schleicher Dialog ... 34
Systemvoraussetzungen ... 30
Interrupt
Beispiel-POEs ... 73
Eingänge X9 / X10 ... 73
mit den Eingängen der XCx 400 / XCx 800, SPS-Beispiel ... 73
Taskstruktur ... 76
UBE 32 0,1I ... 73
Verarbeitung mit dem Funktionsbaustein XFIO_CONFIG ... 73
IP-Adresse
Ändern auf der XCx ... 37
Einstellen/Auslesen auf dem PC ... 36
Einstellen/Auslesen auf der XCx ... 132
überprüfen mit ping ... 36
Java-Applet ... 91
Klemme
X10 – High-Speed Eingänge 5 V DC ... 19
X10, Funktionsauswahl ... 52
X10, X11 schnelle Ausgänge 0-2 / schnelle Eingänge 0-3 ... 65
X11 – High-Speed Ausgänge 5V DC ... 19
X11 Funktionsauswahl ... 52
X9 – Standard I/Os 24V DC ... 19
X9, X10, X11 Bitbelegung ... 65
Koppelspeicher ... 106
Einfügen ... 45
Versionsauswahl ... 47
Zugriff ... 46
LED-Anzeigen
Ethernet ... 22
sercos III ... 22
Steuerung ... 21
Log-Book ... 105
MicroBrowser ... 92
MicroVisIT ... 91
Montage ... 13
MULTIPROG
Bibliotheken und Funktionsbausteine ... 111
Einbindung von ProCANopen ... 86
Installation ... 31
Netzwerkvariablen ... 82
Neues Projekt ... 41
Projekt kompilieren und senden ... 43
Projektbaum ... 41
Multi-Task-System ... 119
Netzgeräte ... 28
Netzwerk-Schnittstelle
Inbetriebnahme ... 35
Kommunikationsschema ... 35
Programmierung mit MULTIPROG ... 38
Netzwerkvariablen ... 82
Online-Funktionen
XFIO ... 50
XRIO ... 50
XUIO ... 50
OPC-Server
Installation ... 32
Version 2.0 ... 96
Version 2.1 ... 97
OPC-Variable
Deklaration ... 95
PDD Process Data Directory ... 92
PDO ... 78
ping ... 36
PNSCommunicating ... 117
Prioritäten der Tasks ... 125
ProCANopen
Einbindung in MULTIPROG ... 86
erste Verbindung ... 87
Installation ... 85
Konfiguration mit ... 84
Konfigurationsmanager auswählen ... 89
POE CANdemo ... 90
Test der Netzwerkverbindung ... 90

Process Data Directory ... 92
Process Data Objects ... 78
Programm, Zuweisen zu Task ... 127
Programmieren ... 103
Programmierung der SPS ... 106
Projektbaum ... 41
PWM, SPS Beispiel ... 73
PWM1/2 mit den schnellen Ausgängen 0/1 ... 72
Reset-Taster ... 23
RS 232 / RS 422 / RS 485-Schnittstellen ... 18
Schleicher Dialog ... 99
  Bedienoberfläche ... 101
  Bedienung der XCx mit dem ... 94
  Installation ... 34
  Konfigurieren ... 100
  Steuerungsmenü ... 102
Schnelle Ausgänge ... 52
schnelle Ausgänge 0/1 als PWM1/2 ... 72
schnelle Ausgänge 0-2 / schnelle Eingänge 0-3 ... 65
Schnelle Eingänge ... 52, 65
Schneller Ausgang ... 65
Schnittstellen
  CAN ... 17
  Ethernet ... 16
  RS 232 / RS 422 / RS 485 ... 18
  sercos III ... 16
  USB 2.0 ... 16
  USB Device ... 17
Schrittmotor ... 52
SDO ... 78
sercos III-Schnittstellen ... 16
Service Data Objects ... 78
Sicherheitshinweise ... 7
Software ... 135
SPGs ... *Siehe* Systemprogramme
SpiderControl PLC Editor ... 91
SPS
  Allgemeines Reset ... 109
  Applikationsbeispiele für XCx400 / XCx800 I/O Schnittstellen ... 65
  Beispiel für Absolutwertgeber ... 70
  Beispiel für Inkrementalwertgeber ... 68
  Beispiel für Interrupt mit den Eingängen der XCx 400 / XCx 800 ... 73
  Beispiel für PWM ... 73
  Betriebszustände ... 107
  Betriebszustände wechseln ... 108
  Grundlagen ... 106
  Programmierung ... 106
  Starten/Stoppen der Programmausführung ... 108
  Systemvariablen ... 110
Standard I/Os 24V DC ... 19
Steckplatzreihenfolge ... 13
Steuereinheit
  Abmessungen ... 26
  Technische Daten ... 24
Steuerungsaufbau ... 12
STOPP Betriebszustand ... 107
Systemprogramme SPGs ... 122
System-Task ... 122
Systemübersicht ... 11
Systemvariablen der SPS ... 110
Systemvoraussetzungen ... 30
Tabelle der RIO-Modultypen ... 64
Task
  Anwender-Task ... 120
  Default-Task ... 124
  Ereignis-Task ... 121
  erstellen in MULTIPROG ... 127
  Prioritäten ... 125
  Prioritätsstufen Übersicht ... 119
  System-Task ... 122
  Task-Information ... 124
  Watchdog ... 126
  zyklische Task ... 120
Task für I/O-Refresh ... 53
TCP/IP-Verbindung ... 30
Technische Daten
  Alle Module ... 134
  Steuereinheit XCx 400 / XCx 800 ... 24
USB 2.0-Schnittstellen ... 16
USB Device Schnittstelle ... 17
Variablen
  Deklaration der OPC-Variablen ... 95
Variablendeklaration
  in den Beispielprogrammen der Funktionsbausteine ... 112
Warenzeichenvermerke ... 135
Watchdog ... 126
Webserver
  aktivieren ... 91
  allgemein ... 91
  Java-Applet ... 91
  MicroBrowser ... 92
  MicroVisIT ... 91
  PDD ... 92
  SpiderControl PLC Editor ... 91
X1, X2 – sercos III-Schnittstellen ... 16
X10 – High-Speed Eingänge 5 V DC ... 19
X11 – High-Speed Ausgänge 5V DC ... 19
X3 – Ethernet-Schnittstellen ... 16
X4 – USB 2.0-Schnittstellen ... 16
X5 – USB Device Schnittstelle ... 17
X6 – CAN-Schnittstellen ... 17
X7 – RS 232 / RS 422 / RS 485-Schnittstellen ... 18
X8 – Betriebsspannung / Betriebsbereitrelais ... 18
X9 – Standard I/Os 24 V DC ... 19
X9, X10, X11 Bitbelegung ... 65
XFIO
  Konfiguration ... 51
  Online-Funktionen ... 50
XRIO ... 53
  Buskoppler hinzufügen ... 54
  Eigenschaften ... 54
  Karte hinzufügen ... 54
  Konfiguration ... 50, 53, 55
  Löschen ... 54
  Modul hinzufügen ... 54
  Online-Funktionen ... 50
  optionale Buskoppler ... 55
  optionale Module ... 55
XUIO
  Eigenschaften ... 58
  Konfiguration ... 50, 58
  Löschen ... 58
  Modul hinzufügen ... 58
  Online-Funktionen ... 50
Zubehör ... 135
Zugangsberechtigung ... 103
Zuweisen, Programm zu Task ... 127
Zyklische Task ... 120